The Dissertation Committee for Dung-Ying Lin Certifies that this is the approved version of the following dissertation:

A Dual Approximation Framework for Dynamic Network Analysis: Congestion Pricing, Traffic Assignment Calibration and Network Design Problem

Committee:

S. Travis Waller, Supervisor

Leon S. Lasdon

Randy B. Machemehl

William J. O'Brien

Zhanmin Zhang

A Dual Approximation Framework for Dynamic Network Analysis: Congestion Pricing, Traffic Assignment Calibration and Network Design Problem

by

Dung-Ying Lin, B.B.A.; M.B.A

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

May, 2009

To My Parents

Acknowledgements

I would like to express my deepest gratitude to my advisor Dr. S. Travis Waller. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own and at the same time the guidance to recover when my steps faltered. His unwavering support during the course of my doctoral study helped me overcome many difficult situations and made this dissertation possible. I would also like to thank the members of my committee: Dr. Leon S. Lasdon, Dr. Randy B. Machemehl, Dr. William J. O'Brien and Dr. Zhanmin Zhang for their insightful suggestions, valuable feedback, continued patience and unfailing understanding. Also, I am sincerely thankful for the help regarding letters of recommendation from Dr. Chandra Bhat and Dr. Ivan Damnjanovic.

I would like to thank the members of the TEQSON lab, Ampol Karoonsoontawong, Avinash Unnikrishnan, David Suescun, Avinash Voruganti, Steve Boyles, Varunraj Valsaraj, Roshan Kumar, Natalia Ruiz Juri, Nezamuddin, Erin Ferguson, Jennifer Duthie, ManWo Ng, Lauren Gardner, Renee Alsup, Deepak Kumar, David Fajardo, Satish Ukkusuri, Chi Xie and Junsik Park, to whom I am indebted for a friendly environment within which to work. Finally, I am grateful for the administrative assistance from Libbie Toler and Chandra Lownes.

Dung-Ying Lin

The University of Texas at Austin

May 2009

A Dual Approximation Framework for Dynamic Network Analysis: Congestion Pricing, Traffic Assignment Calibration and Network Design Problem

Publication No._____

Dung-Ying Lin, Ph.D.

The University of Texas at Austin, 2009

Supervisor: S. Travis Waller

Dynamic Traffic Assignment (DTA) is gaining wider acceptance among agencies and practitioners because it serves as a more realistic representation of real-world traffic phenomena than static traffic assignment. Many metropolitan planning organizations and transportation departments are beginning to utilize DTA to predict traffic flows within their networks when conducting traffic analysis or evaluating management measures. To analyze DTA-based optimization applications, it is critical to obtain the dual (or gradient) information as dual information can typically be employed as a search direction in algorithmic design. However, very limited number of approaches can be used to estimate network-wide dual information while maintaining the potential to scale. This dissertation investigates the theoretical/practical aspects of DTA-based dual approximation techniques and explores DTA applications in the context of various transportation models, such as transportation network design, off-line DTA capacity

calibration and dynamic congestion pricing. Each of the later entities is formulated as bi-level programs.

Transportation Network Design Problem (NDP) aims to determine the optimal network expansion policy under a given budget constraint. NDP is bi-level by nature and can be considered a static case of a Stackelberg game, in which transportation planners (leaders) attempt to optimize the overall transportation system while road users (followers) attempt to achieve their own maximal benefit. The first part of this dissertation attempts to study NDP by combining a decomposition-based algorithmic structure with dual variable approximation techniques derived from linear programming theory.

One of the critical elements in considering any real-time traffic management strategy requires assessing network traffic dynamics. Traffic is inherently dynamic, since it features congestion patterns that evolve over time and queues that form and dissipate over a planning horizon. It is therefore imperative to calibrate the DTA model such that it can accurately reproduce field observations and avoid erroneous flow predictions when evaluating traffic management strategies. Satisfactory calibration of the DTA model is an onerous task due to the large number of variables that can be modified and the intensive computational resources required. In this dissertation, the off-line DTA capacity calibration problem is studied in an attempt to devise a systematic approach for effective model calibration.

Congestion pricing has increasingly been seen as a powerful tool for both managing congestion and generating revenue for infrastructure maintenance and sustainable development. By carefully levying tolls on roadways, a more efficient and optimal network flow pattern can be generated. Furthermore, congestion pricing acts as an effective travel demand management strategy that reduces peak period vehicle trips by

encouraging people to shift to more efficient modes such as transit. Recently, with the increase in the number of highway Build-Operate-Transfer (B-O-T) projects, tolling has been interpreted as an effective way to generate revenue to offset the construction and maintenance costs of infrastructure. To maximize the benefits of congestion pricing, a careful analysis based on dynamic traffic conditions has to be conducted before determining tolls, since sub-optimal tolls can significantly worsen the system performance. Combining a network-wide time-varying toll analysis together with an efficient solution-building approach will be one of the main contributions of this dissertation.

The problems mentioned above are typically framed as bi-level programs, which pose considerable challenges in theory and as well as in application. Due to the non-convex solution space and inherent NP-complete complexity, a majority of recent research efforts have focused on tackling bi-level programs using meta-heuristics. These approaches allow for the efficient exploration of complex solution spaces and the identification of potential global optima. Accordingly, this dissertation also attempts to present and compare several meta-heuristics through extensive numerical experiments to determine the most effective and efficient meta-heuristic, as a means of better investigating realistic network scenarios.

Table of Contents

List of Tables ..... **xiii**
List of Figures ..... **xiv**
Chapter 1. Introduction **1**
1.1. Motivation 1
1.2. Dissertation Contributions 3
1.3. Dissertation Organization 6
Chapter 2. Literature Review **8**
2.1 Network Design Problem 8
2.2 DTA Capacity Calibration 13
2.3 Dynamic Congestion Pricing 15
2.4 Meta-heuristics 18
2.5 Summary 22
Chapter 3. Dual Variable Approximation **23**
3.1 CTM Preliminaries 23
3.2 Notations 25
3.3 CTM-Related Constraints 28
3.4 Re-simulation Dual Approximation Techniques 33
3.5 Summary 35
Chapter 4. Single-destination Bi-level Linear Programming Network Design Problem: A Dantzig-Wolfe Decomposition based Heuristic Scheme **36**
4.1 Primal and Dual Formulations of SONDP 38
4.2 Algorithmic Design 41
4.3 Improved Dual Variables Approximation Techniques 46
4.3.1. Approximation of $\pi_i^{0,t}$ 46
4.3.2. Approximation of $\pi_i^{1,t}, \pi_i^{2,t}, \pi_i^{3,t}$ and $\pi_i^{4,t}$ 51
4.4 Numerical Experiments 54
4.4.1. 6-cell CTM Network 54
4.4.2. 68-cell CTM Network 58
4.5 Summary 63
Chapter 5. Single-destination Bi-level Linear Programming Network Design

Problem: A Dual Approximation Genetic Algorithm 65
5.1 Evaluation Function 68
5.2 Design of the Evaluation Function 68
5.3 Numerical Experiments 69
5.3.1. Experiment1 – 6-Cell CTM Network 70
5.3.2. Experiment 2 – 16-Cell CTM Network 76
5.3.3. Experiment 3 – 68-Cell CTM Network 79
5.4 Summary 79
Chapter 6. Single-destination Off-line Dynamic Traffic Assignment Capacity Calibration 81
6.1 Danzig-Wolfe Based Decomposition Based Heuristic 85
6.2 Dual Variable Approximation 92
6.2.1. Connectivity Algorithm for Dual Variable π_i^0 Approximation 93
6.2.2. Complimentary Slackness Conditions for $\pi_i^{1,t}, \pi_i^{2,t}, \pi_i^{3,t}, \pi_i^{4,t}$ Approximation 96
6.3 Numerical Experiments 100
6.4 Summary 103
Chapter 7. Single-destination Dynamic Congestion Pricing 105
7.1 Solution Methodology: Exact Approach 106
7.2 Solution Methodology: MSA-Based Heuristic 108
7.2.1. Dual Variable Approximation Procedure 108
7.2.2. Approximation of $\pi_i^{0,t}$ 110
7.2.3. Approximation of $\pi_i^{1,t}$, $\pi_i^{2,t}$, $\pi_i^{3,t}$ and $\pi_i^{4,t}$ 112
7.3 Combinatorial Heuristic for UODTA with Tolls 113
7.4 MSA-based Heuristic Overview 115
7.5 Numerical Experiments 117
7.5.1. First-Best Pricing 118
7.5.2. Second-Best Pricing 119
7.5.3. 68-cell CTM Network 121
7.6 Summary 122

Chapter 8. Multiple-destination Bi-level Linear Programming Network Design Problem: A Quantum-Inspired Genetic Algorithm 124
8.1 Algorithmic Design 125
8.2 Direction of Convergence: Rotation Gate 128
8.3 Computational Experiences 130
8.3.1. Sioux Fall Network 131
8.3.2. Sensitivity Analysis of Congestion Level using Sioux Fall Network 135
8.3.3. Sensitivity Analysis of Rotation Gate using Sioux Fall Network 137
8.3.4. Monticello Network 138
8.4 Summary 141
Chapter 9. Multiple-destination Bi-level Linear Programming Network Design Problem: A Descent Method 143
9.1 Mathematical Formulation 143
9.2 A Descent Method 145
9.3 Descent Direction Approximation 147
9.3.1. Approximation of $\frac{\partial F}{\partial b_i}$ 147
9.3.2. Approximation of $\frac{\partial F}{\partial x_{r,s,i}^t}$ 149
9.3.3. Approximation of $\frac{\partial x_{r,s,i}^t}{\partial b_i}$ 149
9.3.4. Approximation of $\frac{\partial F}{\partial y_{r,s,ij}^t}$ 150
9.3.5. Approximation of $\frac{\partial y_{r,s,ij}^t}{\partial b_j}$ 150
9.4 Numerical Experiments 151
9.4.1. Effectiveness 151
9.4.2. Efficiency 152
9.5 Summary 154
Chapter 10. Conclusions and Future Extensions 155

References **160**
Vita **170**

List of Tables

TABLE 1: CHARACTERISTICS OF 6-CELL CTM NETWORK 55

TABLE 2: TIME-DEPENDENT OD DEMANDS FOR 6-CELL CTM NETWORK 55

TABLE 3: 6-CELL CTM NETWORK CELL EXPANSION POLICIES WITH DIFFERENT BUDGET 56

TABLE 4: TIME-DEPENDENT OD DEMAND FOR 68-CELL CTM NETWORK 60

TABLE 5: CHARACTERISTICS OF 68-CELL CTM NETWORK 60

TABLE 6: COMPUTATIONAL TIME FOR GA 62

TABLE 7: CHARACTERISTICS OF 6-CELL CTM NETWORK 71

TABLE 8: TIME-DEPENDENT OD DEMANDS (d_i^t) FOR 6-CELL CTM NETWORK 71

TABLE 9: NUMERICAL RESULTS OF 6-CELL CTM NETWORK WITH 70 VEHICLE TRIPS 72

TABLE 10: NUMERICAL RESULTS OF 6-CELL CTM NETWORK WITH 700 VEHICLE TRIPS 75

TABLE 11: CHARACTERISTICS OF 16-CELL CTM NETWORK 76

TABLE 12: NUMERICAL RESULTS OF 16-CELL CTM NETWORK WITH 45 TIME INTERVALS 78

TABLE 13: NUMERICAL RESULTS OF 16-CELL CTM NETWORK WITH 30 TIME INTERVALS 78

TABLE 14: TIME-DEPENDENT OD DEMANDS FOR 6-CELL CTM NETWORK 100

TABLE 15: CHARACTERISTICS OF 6-CELL CTM NETWORK 100

TABLE 16: CALIBRATED RESULTS 101

TABLE 17: CHARACTERISTICS OF 6-CELL CTM NETWORK 117

TABLE 18: TIME-DEPENDENT OD DEMANDS FOR 6-CELL CTM NETWORK 118

TABLE 19: FIRST-BEST PRICING 118

TABLE 20: SECOND-BEST PRICING 120

TABLE 21: FIRST-BEST PRICING WITH DIFFERENT VOT ($\lambda = \$15.31/hour$) 121

TABLE 22: TSTT FOR DIFFERENT OD DEMAND IN THE 68-CELL CTM NETWORK 122

TABLE 23: $\Delta\theta_i$ FOR ROTATION GATE 129

TABLE 24: INCUMBENT TSTTS WITH DIFFERENT ROTATION GATE ANGLES (HOURS) 137

TABLE 25: 6-CELL CTM NETWORK CELL EXPANSION POLICIES WITH DIFFERENT BUDGETS 152

List of Figures

FIGURE 1: 6-CELL CTM TIME-EXPANDED NETWORK 48
FIGURE 2: 6-CELL CTM NETWORK 54
FIGURE 3: 68-CELL CTM NETWORK 59
FIGURE 4: COMPARISON OF ALGORITHMS PERFORMANCE ($TAB = 100$) 61
FIGURE 5: COMPARISON OF ALGORITHMS PERFORMANCE ($TAB = 50$) 63
FIGURE 6: PROPOSED GENETIC ALGORITHM 66
FIGURE 7: 16-CELL CTM NETWORK 76
FIGURE 8: DANTZIG-WOLFE DECOMPOSITION BASED HEURISTIC 91
FIGURE 9: EXAMPLE OF TIME-EXPANDED NETWORK 93
FIGURE 10: NUMERICAL RESULTS OF THE 68-CELL CTM NETWORK 102
FIGURE 11: UODTA WITH TOLLS 114
FIGURE 12: SIOUX FALL NETWORK 132
FIGURE 13: NUMERICAL RESULTS FOR SIOUX FALL NETWORK WITH VARIOUS BUDGET LEVELS 134
FIGURE 14: NUMERICAL RESULTS FOR SIOUX FALL NETWORK WITH VARIOUS CONGESTION LEVELS 136
FIGURE 15: MONTICELLO NETWORK 139
FIGURE 16: NUMERICAL RESULTS FOR MONTICELLO NETWORK 140
FIGURE 17: 68-CELL CTM NUMERICAL EXPERIMENT 153

Chapter 1. Introduction

1.1. Motivation

Dynamic Traffic Assignment (DTA) is gaining wider acceptance among agencies and practitioners, because it serves as a more realistic representation of real-world traffic phenomena than static traffic assignment. Many metropolitan planning organizations and transportation departments are beginning to utilize DTA to predict traffic flows within their networks when conducting traffic analysis or evaluating management measures. To analyze DTA-based optimization applications, it is critical to obtain dual (or gradient) information, as dual information can typically be employed as a search direction in algorithmic design. However, a very limited number of approaches can be used to estimate network-wide dual information while maintaining scalability. This dissertation investigates the theoretical/practical aspects of DTA-based dual approximation techniques, and explores DTA applications in the context of various transportation models, such as transportation network design, off-line DTA capacity calibration, and dynamic congestion pricing. Each of the latter entities is formulated as a bi-level program.

Transportation Network Design Problem (NDP) aims to determine the optimal network expansion policy under a given budget constraint. NDP is bi-level by nature and can be considered to be a static case of a Stackelberg game, in which transportation planners (leaders) attempt to optimize the overall transportation system while road users (followers) attempt to achieve their own maximal benefit. The first part of this dissertation attempts to study NDP by combining a decomposition-based algorithmic structure with dual variable approximation techniques derived from linear programming theory.

One of the critical elements in considering any real-time traffic management strategy requires the assessment of network traffic dynamics. Traffic is inherently dynamic, since it features congestion patterns that evolve over time and queues that form and dissipate over a planning horizon. It is therefore imperative to calibrate the DTA model such that it can accurately reproduce field observations and avoid erroneous flow predictions when evaluating traffic management strategies. Satisfactory calibration of the DTA model is an onerous task due to the large number of variables that can be modified and the intensive computational resources required. In this dissertation, the off-line DTA capacity calibration problem is studied in an attempt to devise a systematic approach for effective model calibration. By off-line, we assume that transportation planning uses archived traffic data from the past to calibrate the DTA model instead of using real-time data transmitted from an on-site surveillance system.

Congestion pricing has increasingly been seen as a powerful tool for both managing congestion and generating revenue for infrastructure maintenance and sustainable development. By carefully levying tolls on roadways, a more efficient and optimal network flow pattern can be generated. Furthermore, congestion pricing acts as an effective travel demand management strategy; it reduces peak period vehicle trips by encouraging people to shift to more efficient modes, such as transit. With the recent increase in the number of highway Build-Operate-Transfer (B-O-T) projects, tolling has been seen as an effective way to generate revenue and offset the construction/ maintenance costs of infrastructure. To maximize the benefits of time-varying congestion pricing, a careful analysis based on dynamic traffic conditions has to be conducted before determining tolls, since sub-optimal tolls can significantly worsen the system performance. Combining a network-wide time-varying toll analysis with an

efficient solution-building approach will be one of the main contributions of this dissertation.

The problems mentioned above are typically framed as bi-level programs, which pose considerable challenges in theory and in application. Due to the non-convex solution space and inherent NP-complete complexity, the majority of recent research efforts have focused on tackling bi-level programs using meta-heuristics. These approaches allow for efficient exploration of complex solution spaces and the identification of potential global optima. Accordingly, this dissertation also attempts to present and compare several meta-heuristics through extensive numerical experiments to determine effective and efficient meta-heuristics as a means to improve the investigation of realistic network scenarios.

1.2. Dissertation Contributions

Transportation planning problems can typically be characterized as bi-level programs that include transportation planners' decisions in the upper-level programs and road users' responses in the lower-level programs. In this dissertation, we primarily consider three such problems: the Network Design Problem (NDP), Dynamic Traffic Assignment (DTA) calibration, and dynamic congestion pricing.

Transportation network improvements can account for a substantial portion of a given transportation budget and can be naturally framed as a bi-level program that determines the optimal network capacity expansion policy under a budget constraint. This stream of resource planning problems is known as NDP. NDP has been shown to be an NP-complete problem (Johnson et al., 1978)—numerous technical challenges have thwarted efforts to solve it using traditional mathematical programming techniques. Depending on the nature of the traffic assignment model, NDPs can generally be classified into static and dynamic transportation network design problems. The dynamic

network design model is increasingly preferred over static network design approaches, due its ability to reproduce more realistic traffic phenomena. The inability of static models to represent critical traffic behaviors (i.e., bottlenecking, shockwave propagation, and link spill-over) has been well-documented. For instance, Janson (1995) investigates dynamic network design alongside static modeling, as does Waller (2000).

First and foremost, static models do not accurately account for bottlenecks, which have major implications for capacity expansions. For instance, if one considers three consecutive identical links and examines a capacity expansion to the middle link, a static model might predict a substantial decrease in the travel time over the three links, while a dynamic model might predict absolutely no improvement regardless of how much capacity is added to the middle link. Therefore, we exclusively consider the dynamic network design problem in this dissertation. To be precise, this dissertation is concerned with bi-level dynamic NDPs with continuous decision variables, in contexts where continuous investment allows for the possibility of fractional lane additions.

Many Metropolitan Planning Organizations (MPOs) and Departments of Transportation (DOTs) are beginning to utilize DTA to predict network flows when conducting traffic analysis or evaluating management measures. Thus it is imperative to calibrate a DTA model so that the calibrated model can accurately reproduce the field observations and avoid erroneous network flow predictions. However, calibrating a DTA model is onerous, due to the large number of variables that can be modified and the significant computational resources that are often required to perform DTA. If a DTA model is simply calibrated using a trial-and-error approach, the time and effort required can potentially be prohibitive, which points to the need for a systematic DTA calibration procedure. In general, the difference between actual field observations and DTA predictions has been attributed to the following four causes: (1) variability in user

behavior, (2) incorrect trip table, (3) errors in traffic counts, and (4) erroneous capacity values. While each of the four aforementioned issues is highly critical, relatively little research has been conducted regarding capacity calibration. In fact, calibrating capacity values to match observed traffic counts is a relatively common practice in static traffic assignment scenarios. The justification for this process is that numerous operational realities exist at specific facilities that are not observed by the planning agency, due to the regional scope of most planning models. Therefore, there exists a broad range of acceptable capacity values, as the appropriateness of the capacity value is most practically resolved by the ability of the model to match observed traffic counts. Furthermore, with regard to DTA, it is likely that any existing planning capacity values, which may well have been calibrated to match static traffic assignment flows, would be inappropriate for DTA use. This is because dynamic models represent traffic in fundamentally different ways than static approaches do (for instance, many link capacities are modified to take into account the impacts of traffic signals in static approaches).

Numerous studies have been conducted to examine modifications to trip tables. While arguments exist that trip table estimation may be the most critical problem, it is clear that an optimal approach should also consider demand and supply calibration. Therefore, this dissertation examines the later in order to facilitate the eventual development of multi-variable system-wide calibration methodologies, which refer to methods that consider demand, supply, erroneous traffic counts, and non-standard behavioral assumptions; such methods are currently beyond the scope of any single effort.

Congestion pricing is one of the most powerful policy tools used by government agencies as a mean of reducing congestion and encouraging alternative travel modes.

Carefully levied toll strategies regulate travel demand, and make it possible to manage congestion without increasing infrastructure supply. Furthermore, congestion pricing can reduce peak period vehicle trips by encouraging people to shift to more efficient modes. To maximize the benefits of congestion pricing, a careful analysis based on dynamic traffic conditions must be conducted before determining tolls, as sub-optimal tolls can have a significant negative impact on the system's performance. Numerous studies have been conducted to explore innovative techniques that account for various factors, including elasticity of demand, value of time, types of tolling, etc. This dissertation contributes to the growing body of congestion pricing literature by investigating a heuristic but practical approach to determine time-varying tolls in networks with fixed demand.

The first goal of this dissertation is to present mathematical formulations to characterize these problems, so that tailored solution techniques can be developed to streamline the transportation planning process. The second primary focus of this dissertation is to devise novel dual approximation techniques that can be embedded into various solution schemes to efficiently tackle complex transportation problems. The third contribution is the scalable design of solution strategies that account for traffic dynamics and user responses. We conduct numerical experiments on networks of various sizes and topologies to validate our work and derive other insights.

1.3. Dissertation Organization

This dissertation is organized into ten chapters. The first chapter presents an overview of the topics, challenges, and solution methodologies covered in this dissertation. Chapter 2 reviews the related research. Chapter 3 introduces the core concept of dual variable approximation, which is employed throughout this dissertation. Chapters 4 and 5 study the problem of single-destination bi-level dynamic network

design problem. To solve this problem efficiently, we propose a decomposition-based heuristic and a meta-heuristic in Chapter 4 and Chapter 5 respectively. Off-line dynamic traffic assignment calibration problem is analyzed and tackled in Chapter 6. Chapter 7 investigates the dynamic congestion pricing problem and presents a method of successive average to solve it. Chapter 8 presents a quantum-inspired genetic algorithm to obtain near-optimal solutions to multiple-destination bi-level dynamic network design problems in larger networks. Chapter 9 extends the solution framework proposed in Chapter 4 and presents a descent method that incorporates the dual approximation techniques to solve multiple-destination bi-level dynamic network design problems. Numerical experiments, analyses, and insights are presented and discussed in the corresponding chapters. The last section concludes the dissertation and proposes potential future extensions.

Chapter 2. Literature Review

In this chapter, we review the literature concerning NDP, DTA calibration, and dynamic congestion pricing. We critically analyze the key features and assumptions of various models, algorithmic designs, and analytical implications.

2.1 Network Design Problem

The network design problem is at the core of many planning problems and has been extensively studied in the literature. Magnanti and Wong (1984), Boyce (1984), Friesz (1985), Yang and Bell (1998), and Karoonsoontawong (2006) have conducted comprehensive reviews of traffic assignment-based NDPs. This section primarily offers an updated comprehensive review. Typically, the NDP objective function minimizes the Total System Travel Time (TSTT), subject to flow conditions and budget constraints, or it minimizes TSTT plus the cost converted to equivalent time units in the context of applicable flow conditions. The NDP formulations can be classified according to four criteria (Karoonsoontawong, 2006):

(1) User behavior assumptions (i.e., System-Optimal (SO) or User-Optimal (UO)). SO behavior is mathematically tractable but relatively unrealistic, while UO behavior typically compounds the underlying problem but is generally more realistic.

(2) Time resolution (i.e., static or dynamic traffic assignment). The static traffic assignment protocol assumes steady-state conditions, while dynamic traffic assignment accounts for time dynamics.

(3) Decision variables (i.e., discrete or continuous investment variables). The discrete investment variable allows only the addition of entire lanes or new links, while continuous investment permits the addition of fractions of lanes. Since most roads in urban areas were constructed many years ago, discrete variables may be practical only in

certain limited cases. On the other hand, the continuous investment variable has been extensively employed in the literature, and is more justifiable. Essentially, continuous link expansion can be implemented by altering lane widths, median placement, and shoulder areas, which makes continuous NDPs a practical approach to network improvement.

(4) Parameter properties (i.e., deterministic or stochastic parameters). Problem parameters have typically been considered deterministic as opposed to stochastic.

The NDPs based on SO static traffic assignment and deterministic parameters have been extensively studied (for examples, see LeBlanc (1975), Hoang (1982), LeBlanc and Abdulaal (1979), Dantzig et al. (1979), and LeBlanc and Abdulaal (1984)). LeBlanc (1975) formulated the problem as a Mixed Integer Nonlinear Programming (MINLP) model with discrete investment variables, and suggested a branch-and-bound algorithm to find a solution. The proposed solution procedure utilized SO traffic assignment as a lower bound to effectively prune the tree nodes of a branch-and-bound tree and to prevent the use of invalid lower bounds that can create problems in light of Braess' paradox. Hoang (1982) proposed the generalized Benders' partition method to solve the MINLP model. LeBlanc and Abdulaal (1979) proposed a computationally efficient dual approach to tackle the SO NDP with continuous investment variables. Dantzig et al. (1979) presented a decomposition algorithm that segments the NDP into separate subproblems for links. A Lagrangian-based iterative search procedure that relaxes the budget constraint was then devised to solve large-scale SO NDPs. LeBlanc and Abdulaal (1984) compared the approximate representations of the continuous UO and SO NDP models, and showed that SO NDP solutions were as good as those from the UO NDP.

The literature on UO static traffic assignment and deterministic parameters includes publications by Abdulaal and LeBlanc (1979), Suwansirikul et al. (1987), Marcotte (1983), Meng et al. (2001), Friesz et al. (1992), LeBlanc and Boyce (1986), and Patriksson and Rockafellar (2002). Abdulaal and LeBlanc (1979) formulated the continuous NDP as an unconstrained NLP model and showed that the Hooke-Jeeves algorithm (Hooke and Jeeves, 1961) outperformed Powell's method (Powell, 1964). Suwansirikul et al. (1987) showed that a heuristic, equilibrium-decomposed optimization was computationally more efficient than the Hooke-Jeeves algorithm. Marcotte (1983) suggested a constraint accumulation algorithm and proposed iterative optimization assignments, the latter of which are efficient but may not converge to an optimal solution. Meng et al. (2001) contributed an equivalent continuously differentiable non-convex model, and identified an exact local solution by applying an efficient, convergent, augmented Lagrangian method. Friesz et al. (1992) proposed a simulated annealing approach to solve the continuous non-convex NLP model, given variational inequality constraints. This method is computationally expensive, but it yields solutions that are close to the global optimum. LeBlanc and Boyce (1986) re-formulated the continuous model as a bi-level linear program and suggested the efficient point algorithm by Bard (1983) coupled with the Frank-Wolfe approximation for large networks. Later, Marcotte (1988) proved, via a counterexample, that Bard's efficient point algorithm is incapable of identifying the true optimum in some cases. However, the bi-level program remains valid, and other existing algorithms that guarantee the exact solution may be similarly applicable. Patriksson and Rockafellar (2002) showed that a constrained local Lipschitz minimization problem is equivalent to the Mathematical Program with Equilibrium Constraints (MPEC), and they suggested a descent-type algorithm for the MPEC.

Stochastic User Optimal (SUO) assignment is based on the assumption that an individual's perceived travel costs are subject to random error. The literature on NDP SUO assignments includes contributions from Davis (1994), Chen and Alfa (1991), and Uchida et al. (2008). Davis (1994) showed that a continuous NDP with logit-based SUO assignment leads to a differentiable, large-scale, and tractable problem and suggested a procedure to calculate the derivatives of the SUO assignment without computing the route choice probabilities. Later, Yang and Bell (1998) pointed out that this model may yield unreasonable results, because it generally overestimates traffic flow on overlapping routes. Chen and Alfa (1991) solved the discrete model with a heuristic based on the branch-and-bound method and the logit-based incremental traffic assignment algorithm. Uchida et al. (2008) studied the multi-modal network design problem in the context of a probit-based stochastic user equilibrium. The railway, bus, and automobile were simultaneously considered as a means of capturing interaction effects.

The abovementioned static NDP models present three drawbacks when compared to the DTA-based models. First, the static models cannot capture the traffic interactions among adjacent links. In contrast, the DTA-based model captures traffic propagation among adjacent links with a greater amount of fidelity. Second, the static models assume steady-state time-invariant Origin-Destination (OD) demand; this is obviously unrealistic during the peak period and may lead to suboptimal solutions. The DTA-based models overcome this deficiency by capturing dynamic OD demand and by providing recommendations that optimize the network for the duration of the assignment. Third, in the static models, any capacity expansions typically appear in the denominator of the link performance function, which creates nonlinearities that are difficult to handle in a mathematical program. In the DTA-based model, the capacity expansions for each

road segment can be captured by a single parameter on the right-hand side of a linear constraint, a notion that will be investigated in later sections.

There is an increasing body of literature on the DTA-based NDP. Janson (1995) and Waller (2000) showed that the DTA-based NDP model is more promising than the static model. Relevant DTA-based NDP models include those based on the single-destination System-Optimal Dynamic Traffic Assignment (SODTA) and User-Optimal Dynamic Traffic Assignment (UODTA) Linear Programming (LP) models with fixed-departure-time OD demands that were introduced by Ziliaskopoulos (2000) and Ukkusuri (2002), respectively. Both SODTA and UODTA LP models propagate traffic according to the CTM, a traffic flow theoretical model by Daganzo (1994 and 1995). The DTA-based NDP models can be further classified into two categories: (1) single-level models and (2) bi-level models. The single-level models include those proposed by Waller and Ziliaskopoulos (2001), Ukkusuri and Waller (2008), Ukkusuri et al. (2004), Karoonsoontawong and Waller (2005), and Waller et al. (2006). The bi-level models include contributions from Jeon et al. (2005) and Karoonsoontawong and Waller (2006, 2007 and 2008).

In regard to single-level models, Waller et al. (2006) and Ukkusuri and Waller (2008) formulated the continuous SO and UO NDP LP models. Furthermore, Waller and Ziliaskopoulos (2001) introduced the stochastic SODTA-based NDP with long term OD demand uncertainty, formulated as a two-stage Stochastic Linear Program with Recourse (SLPR) and a Chance-Constrained Program (CCP). Ukkusuri et al. (2004) introduced UO versions of the SLPR and CCP models. Karoonsoontawong and Waller (2005) conducted a comprehensive comparison of the SO and UO SLPR models. Turning to bi-level models, Jeon et al. (2005) formulated a bi-level UODTA mixed-integer programming model and employed a genetic algorithm to solve the

discrete NDP by allowing either one or zero lane additions. Karoonsoontawong and Waller (2006) proposed the UODTA-based NDP linear bi-level programming formulation with exact solution methods, and furthermore, they developed three meta-heuristics (simulated annealing, genetic algorithm, and random search) for the larger-size multi-origin multi-destination problem. Karoonsoontawong and Waller (2007) further formulated the robust dynamic network design problem as a quadratic-linear bi-level program and proposed an exact solution method. Signal setting and lane layout for urban networks were studied by Cantarella et al. (2006). Karoonsoontawong and Waller (2008) proposed a robust optimization model for the combined NDP, UODTA, and traffic signal setting design. Yin et al. (2008) proposed three optimization models (sensitivity-based, scenario-based, and min-max) to solve the robust network design problem with demand uncertainty.

2.2 DTA Capacity Calibration

Generally, the differences between actual field observations and DTA predictions have been attributed to the following four causes:

(1) Variability in user behavior: in dynamic traffic assignment scenarios, users are often assumed to be rational and have perfect information. However, user behaviors can be diverse and difficult to predict. This issue has been specifically addressed by considering the heterogeneity or stochasticity of user behaviors (for example, see Peeta and Yu (2006), Mahmassani et al. (2005), and Cascetta and Cantarella (1991)).

(2) Incorrect trip table: the trip table is usually assumed to be exogenous to dynamic traffic assignment. Hence, an incorrect trip table generally leads to erroneous assignments. Different approaches have been proposed to identify realistic trip tables (for example, see Cascetta and Postorino (2001), Sherali and Park (2001), and Sherali et al. (1997)).

(3) Errors in traffic counts: the counts available may be incorrect due to an improper data collection process. Other researchers have approached this issue in various ways (for example, see Chen and May (1987), Turner et al. (2000), and Chen et al. (2003)).

(4) Erroneous capacity values: in dynamic traffic assignment scenarios, network capacities are assumed to be predetermined and exogenous. However, this is a strong assumption that may not be true in practice. For instance, the capacities in certain links may be lower than the designed values due to roadside parking, or to other issues that planning agencies do not record. Despite its obvious importance in the context of capacity calibration, relatively little research has been conducted to deal with this issue. To the best of our knowledge, Kunde (2002) is the only published work that exclusively calibrates the capacity of the DynaMIT-P simulation module, using the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm. This dissertation contributes to the capacity calibration literature by analyzing problem features and consequently devising a systematic approach to tackle this problem.

Four major DTA solution and formulation methodologies have been reviewed by Peeta and Ziliaskopoulos (2001): mathematical programming, variational inequality, optimal control, and simulation-based approaches. However, due to the difficulty of conducting large-scale experiments with analytical approaches, simulation-based explorations are commonly used in practice and will be the focus of the following review. The calibration of simulation models in operations research was studied by Kleijnen (1995). Both model verification and validation were surveyed, which makes this publication a general framework for the calibration of traffic simulations. Jha et al. (2004) offered insight into the practical challenges encountered when applying and calibrating large-scale simulations, including data collection, computational requirements/problem sizes, and conversion of planning Origin-Destination (OD) data to

simulation OD parameters. Ishak et al. (2006) conducted extensive experiments to demonstrate the advantages of both the calibration and the computation associated with the Cell Transmission Model (CTM), proposed by Daganzo (1994 and 1995). He and Ran (2000) derived a set of likelihood functions to calibrate and validate dynamic route choice and flow propagation of dynamic traffic assignment. Muloz et al. (2004) developed a semi-automated method to calibrate the CTM. The proposed method employs a least-squares data fitting approach to calibrate free-flow speeds, shockwave speeds, and jam densities for a single freeway segment. Mahut et al. (2004) exploited a simulation-based calibration model that adjusted various parameters, such as the gap-acceptance and average vehicle length, to match one-hour empirical traffic counts. The authors also presented the first attempt to simultaneously calibrate path and link flows using a simulation-based DTA model. Balakrishna et al. (2007) presented an off-line DTA calibration model that simultaneously estimates the transportation demand-side and supply-side parameters. Their calibration problem is formulated as a complex stochastic mathematical program with a high degree of nonlinearity. Gradient-based Simultaneous Perturbation Stochastic Approximation (SPSA) was employed to tackle the problem.

2.3 Dynamic Congestion Pricing

Traditionally, optimal tolls have been calculated based on Pigouvian taxes (Pigou, 1920), wherein every individual is charged a toll equivalent to the negative externality that the traveler imposes on the other users of the system. Such tolls are termed marginal social cost prices as they maximize the social welfare or system performance (Arnott and Small, 1994; Knight, 1924; Walters, 1961; Morrison, 1986). Outside of maximizing social welfare, numerous other models and solution algorithms have also been developed to arrive at the optimal first-best or second-best tolls to optimize other

objective functions, including the revenues generated, maximum tolls collected, etc. (Hearn and Yildirim, 2001; Yildirim and Hearn, 2005; Yang and Lam, 1996; Labbe, Marcotte and Savard, 1998; Ferrari, 2002; Patriksson and Rockafellar, 2002; Verhoef, 2002; Lawphongpanich and Hearn, 2004). However, all of the above works are static in nature and do not account for time-varying congestion patterns in the context of optimal toll identification.

Vickrey (1969) used the bottleneck model to determine time-varying tolls for the purposes of eliminating queuing congestion on a single link. The single bottleneck model has been extended, and various other issues, such as unobserved heterogeneity in user valuations of travel time and variations in link exit capacities, have been tackled by numerous authors that include Arnott et al. (1990), Arnott and Kraus (1998), and Yang and Huang (1997). Braid (1996) and DePalma and Lindsey (2000) applied the bottleneck model to determine optimal tolls on competing routes connecting a single origin-destination pair. Despite providing numerous invaluable insights, one of the common deficiencies of the bottleneck models is its premise that traffic is either in free-flow or at zero speed when waiting in the queue. Another approach, first proposed by Henderson (1974 and 1981) and later developed by Chu (1995), offered a method to determine time-varying tolls while incorporating speed as a function of the degree of network congestion. However, the abovementioned publications only determine analytical relationships for time-varying tolls on simplified single- or double-link networks and do not consider the spatial evolution of traffic dynamics on generalized networks.

Carey and Srinivasan (1993) derived approximate analytical expressions for the externality imposed by an individual on other users, and hence the congestion tolls on general networks, using the Kuhn-Tucker optimality conditions. Wie and Tobin (1998)

developed a convex optimal control formulation to determine destination-based first-best dynamic marginal tolls for generalized networks. However, the aforementioned publication assumes that all the links in the network can be priced. More recently, Joksimovic et al. (2005) presented a mathematical program that used an equilibrium constraints formulation to determine the optimal uniform and time-varying tolls. A simple iterative grid search approach was proposed by the author. For the road pricing problem, all combinations were explored explicitly, which limited the applicability of the proposed method to real-world networks.

Wie (2007) provided a bi-level formulation to determine triangular-shaped multi-step congestion tolls in the context of general networks to maximize consumer surplus. Analytical relationships were used to determine arc travel times on the basis of free flow speeds, flow rates, and traffic volumes; however, this approach likely failed to capture dynamic congestion phenomena like shockwaves. De Palma (2005) conducted a simulation-based analysis to determine the impact of six types of link tolling schemes, including flat tolls, second-best cordon tolls, etc. Approximate heuristic methods based on fitting a quadratic response surface were developed to determine near-optimal tolls.

As can be seen from the reviewed literature, bi-level programs generally result in reasonable models for formulating traffic management measures and analyzing planning strategies. Numerous research efforts have focused on bi-level programming in recent years. Vicente and Calamai (1994) and Colson et al. (2007) gave extensive surveys of the existing solution approaches in the realm of bi-level programming, including extreme point algorithms, branch-and-bound algorithms, complementary pivot algorithms, descent methods, penalty function methods, and trust region methods. However, due to the proven NP-complete/NP-hard nature of bi-level programming, Colson et al. (2007)

suggested that exploring the combinatorial structure of such problems will likely lead to good solution approaches, which is essentially the strategy adopted in this dissertation.

2.4 Meta-heuristics

The network design problem (NDP) based on dynamic traffic assignment is NP-complete. Karoonsoontawong (2006) conducted comprehensive reviews of DTA-based NDPs together with static traffic assignment-based NDPs. Among the existing methods for solving DTA-based NDPs, analytical approaches and meta-heuristics are the primary candidates. To efficiently solve this bi-level programming problem, Lin et al. (2009) developed a Dantzig-Wolfe decomposition-based heuristic scheme that utilizes the backward connectivity algorithm and complimentary slackness conditions in approximating the dual variables required for the decomposition framework. However, the decomposition scheme is devised specifically for single-destination problems. Karoonsoontawong and Waller (2006) developed a modified K^{th}-best algorithm and a mixed integer programming reformulation technique to solve the bi-level dynamic network design problem. In addition, Colson et al. (2007) provided a detailed overview of bi-level optimization techniques, all of which can be modified to solve dynamic network design problems. However, even though the abovementioned analytic methods can potentially find optimal solutions to the problem, analytical methods are typically not scalable and have limited practical applicability. Consequently, the majority of the research efforts have been focused on tackling this problem using meta-heuristics, especially when large-scale problems are of interest.

Among the existing meta-heuristics, the Genetic Algorithm (GA) is one of the more commonly used techniques. GA, introduced by Holland (1975), is recognized as an effective search procedure for optimization problems with complex search spaces. GA is an iterative procedure that maintains a population of candidate solutions to the

objective function. The population usually starts as a group of randomly generated candidate solutions. During each generation, the current population is evaluated, and fitness values are obtained for each of the candidate solutions. A new population is stochastically generated using crossover, mutation, and selection operations on the basis of these fitness values. The new population is used in the following iteration of the algorithm. This procedure repeats until a stopping criterion is met. Numerous applications of GAs have demonstrated their impressive efficiency in practice.

In solving the dynamic transportation NDP, Karoonsoontawong and Waller (2006) applied Simulated Annealing (SA), GA, and Random Search (RS) to solve the bi-level dynamic network design problem. According to the preliminary results presented in that work, GA outperforms the other meta-heuristics for test networks of various sizes. Jeon et al. (2006) employed a Selectorecombinative Genetic Algorithm to solve a discrete bi-level dynamic network design problem in which only one lane addition is allowed when the candidate link is selected for capacity expansion. Lin et al. (2008) incorporated the dual variable approximation techniques into the paradigm of genetic algorithms and devised an efficient evaluation function using these approximation techniques. Though the dynamic network design problem can be solved by the proposed genetic algorithm with great efficiency, their approaches were developed specifically for single-destination problems.

From the surveyed literature, it can be seen that multiple functional evaluations are inevitable when solving a dynamic network design problem with meta-heuristics. However, with the conventional evaluation function for Dynamic Traffic Assignment (DTA), the functional evaluation is computationally expensive and can be prohibitive. For instance, it requires more than 48 hours for the DTA to converge when studying the city of Austin, Texas using simulation-based DTA modules. If the scale of

computational effort cannot be reduced, it will be impractical to employ meta-heuristics in tackling this problem. To address this issue, there exist two possible strategies: 1) Reduce the computational effort in each functional evaluation (perhaps by replacing the DTA evaluation with other more efficient, though approximate, evaluation functions); 2) Reduce the number of functional evaluations. In this dissertation, we develop a new evaluation function to replace the DTA traffic simulation, following the first strategy. Aside from the first strategy, we also attempt to follow the second strategy by employing a Quantum-inspired Genetic Algorithm (QGA) on a classical computer so that more information can be contained within one qubit chromosome and hence reduce the number of necessary functional evaluations. The following paragraphs provide a review of quantum computing literature.

Classical computers use patterns of grouped bits to represent numbers. Each bit can hold either a zero or one that correspond to an off or on state. Thus, a classical computer can represent only one out of the possible 2^n states at any instant when n bits are used. In the early 1980s, the idea of a quantum computer was first proposed by Benioff (1980), and it addressed this inefficiency in state representation. Quantum computation has garnered significant attention since then. A quantum computer, unlike a classical computer, utilizes the qubit to represent different system states. A qubit can hold the number one, zero or any superposition of the two numbers at the atomic level. Hence, 2^n states can be represented by only n qubits in a quantum computer instead of the 2^n bits required in a classical computer.

The concept of representing data in qubit form can be exploited to design novel algorithms, with complexities that far surpass the power of conventional algorithms. For instance, one of the most important quantum algorithms in recent years is the factorization algorithm by Shor (1994 and 1999). Shor's algorithm factors an integer in

polynomial time using modular exponentiation and the Quantum Fourier Transform. Since the widely used RSA computer encryption scheme relies on the assumption that factorization of a large number is computationally infeasible with classical computers, Shor's algorithm can potentially break RSA encryption with quantum computers. Tony (1999) gave an example to demonstrate the computational power of Shor's factorization algorithm: the time required to factor a 129 digit number (known as *RSA129*) using 1,000 classical workstations is 8 months, while the time required to factor the same number with Shor's algorithm using a 100 MHz quantum computer is a few seconds. Another important example is the database search algorithm by Grover (1996). The proposed quantum algorithm searches N random ordered items in $O(\sqrt{N})$ steps instead of the well-known $N/2$ steps for a classical computer. These pioneering research efforts suggest that it is possible to solve problems that are traditionally regarded as NP-hard/NP-complete in polynomial time by using a quantum computer. In addition to the abovementioned research, a growing body of literature on quantum computation can be found. DiVincenzo (1995) provided an excellent overview of quantum computation and also points out the difficulties in designing quantum computers. Steane (1998) provided a comprehensive survey of quantum computing and quantum information theory after the 1980s.

Research on integrating a genetic algorithm and quantum computing can be broadly classified into two major areas (Giraldi et al., 2004): 1) New quantum algorithm designs that take advantage of both the genetic algorithm and quantum computing parallelism; 2) Develop quantum-inspired genetic algorithms that employ quantum mechanical principles to search for the optimal solution. Giraldi et al. (2004) provided a survey of the main research efforts in both areas. Due to the technical difficulties of building quantum computers, it is currently difficult to experiment with

algorithms on quantum computers. Only a few advanced research laboratories, such as IBM-Almaden Research Center (2000), are capable of building quantum computers. Therefore, the second approach is more appealing for engineering practice, since it can be implemented on classical 0-1 computers. One of the most successful applications is the work by Han and Kim (2002). They developed a quantum-inspired evolutionary algorithm to solve the knapsack problem, and demonstrated the effectiveness and applicability of quantum-inspired algorithms. Computational issues together with practical implementation have been extensively discussed in Han et al. (2001), Han and Kim (2002), and Han and Kim (2003). Note that the algorithm developed in their work is suitable for problems with binary decision variables.

2.5 Summary

This chapter overviews the literature relevant to the research conducted in this dissertation and forms the foundation for it. Despite of significant amount of work focused on development of DTA-based optimization model, there has been relatively little work conducted in facilitating the linear programming structure of DTA-based optimization problem and exploring the system-wise gradient information in designing solution algorithms. Moreover, scalability of solution algorithms for DTA-based optimization problems has not been well-studied in the literature. Thus, calculating system-wise gradient and exploring scalable solution framework are essentially the two major goals of this dissertation. The next chapter describes the cell transmission model embedded in the DTA-based optimization problems and proposes the dual variable approximation techniques that are more accurate but may be impractical in realistic problems.

Chapter 3. Dual Variable Approximation

To begin, we briefly review the Cell Transmission Model (CTM) embedded in the DTA-based problems considered in this dissertation. We then examine the mathematical formulations that encapsulate CTM-related constraints to capture traffic dynamics and user behaviors. Finally, we propose potential solution techniques together with a set of dual variable approximation techniques consistent with our proposed problem formulations.

3.1 CTM Preliminaries

The CTM, as devised by Daganzo (1994 and 1995), is a discrete approximation of the continuum traffic flow model that can accurately describe traffic dynamics according to hydrodynamic theory (Lighthill and Whitham (1955), Richards (1956)). CTM works by converting the network into a series of cells connected by cell connectors. During any time interval t, there are three parameters associated with a cell i:

(1) jam density N_i^t: the maximum number of vehicles allowed in cell *i* during time interval *t*.

(2) saturation flow rate Q_i^t: the maximum number of vehicles that can flow into or out of cell *i* during time interval *t*

(3) δ_i^t: the ratio of free flow speed to backward propagation speed for each cell *i* and time interval *t*.

The state of the system is defined by the number of vehicles in cell *i* during a time interval *t*, denoted by x_i^t, which is obtained with the simple mathematical relationships below and updated at every time step:

$$x_i^t = x_i^{t-1} + y_{i-1,i}^{t-1} - y_{i,i+1}^{t-1}$$

For cells in normal freeway segments without entrance and exit ramps, the equation indicates that x_i^t can be obtained from $y_{i-1,i}^{t-1}$ (the number of vehicles moving from cell *i-1* to cell *i* during time interval *t-1*) and $y_{i,i+1}^{t-1}$ (the number of vehicles moving from cell *i* to cell *i+1* during time interval *t-1*).

$y_{i,i+1}^{t-1}$ is calculated from the minimum of the maximum number of vehicles that can flow from the upstream cell (S_i^{t-1}) and the maximum number of vehicles that can be received by the downstream cell (R_{i+1}^{t-1}). S_i^{t-1} and R_i^{t-1} are computed using the following equations:

$$S_i^{t-1} = \min\{x_i^{t-1}, Q_i^{t-1}\}$$

$$R_{i+1}^{t-1} = \min\{Q_{i+1}^{t-1}, \delta_i^t (N_{i+1}^{t-1} - x_{i+1}^{t-1})\}$$

When using CTM to simulate traffic propagation, special attention needs to be paid to the choice of simulation interval and the length of each cell. The simulation interval used in this dissertation is 6 seconds (commonly employed in practice), which allows for an adequate representation of the traffic dynamics. The length of a cell must be greater than or equal to the minimum distance traveled in a simulation interval under free flow conditions. For instance, if the free-flow speed of a roadway segment is 60 mph (316,800 feet/hour), the minimum cell length is 528 feet ($\frac{316{,}800}{60 \times 60} \times 6$) in that

segment. It is also worth noting that the cell length employed in this dissertation will be fixed; variable cell length can be used for some specific applications.

3.2 Notations

For future convenience, we present the notations (Sets, Parameters and Variables) used throughout this dissertation.

Sets

C = ordinary cells

C_S = sink cells

C_R = source cells

T = discrete time intervals

E = ordinary cell connectors

E_S = sink cell connectors

$FS(i)$ = cell connectors emanating from cell i

$RS(i)$ = cell connectors emanating to cell i

Parameters

TAB = total available budget

δ_i^t = ratio of link free flow speed to backward propagation speed for each cell and time interval

M_t = cost per time interval that yields user-optimal flows

ζ_i = initial number of vehicles in cell *i*

d_i^t = single-destination deterministic demand originating from cell *i* during time interval *t*

$d_{r,s}^t$ = multiple-destination deterministic demand originating from source *r* and destining to destination *s* during time interval *t*

N_i^t = maximum number of vehicles allowed in cell *i* during time interval *t*

Q_i^t = maximum number of vehicles that can flow into or out of cell *i* during time interval *t*

χ_i = increase in N_i^t per unit of budget b_i invested

ϕ_i = increase in Q_i^t per unit of budget b_i invested

$x_{i,a}^t$ = actual number of vehicles observed in the field in cell *i* during time interval *t*

$\chi_{i,\min}, \chi_{i,\max}$ = min and max perturbations of the jam density of cell *i*

$\phi_{i,\min}, \phi_{i,\max}$ = min and max perturbations of the saturation flow rate of cell *i*

Variables

b_i = budget allocated to cell *i*

x_i^t = number of vehicles in cell *i* during time interval *t*

y_{ij}^t = number of vehicles moving from cell *i* to cell *j* during time interval *t*

$\pi_i^{0,t}$ - $\pi_i^{4,t}$ = dual variables of the corresponding constraints

$\rho_i^{1,t}$ - $\rho_i^{2,t}$, ρ_i^3 - ρ_i^6 = dual variables of the corresponding constraints in off-line DTA capacity calibration formulation

$\hat{\chi}_i$ = perturbation of the jam density of cell *i*

$\hat{\phi}_i$ = perturbation of the saturation flow rate of cell *i*

z_i^t = variable introduced for cell *i* during time interval *t*, to reformulate the absolute-value objective function of the off-line DTA capacity calibration formulation

ω_i^t = time-dependent tolls in cell *i* during time interval *t*

ω = vector of time-dependent tolls

Ξ = vector of path flows to represent any feasible DTA

Ξ^* = vector of path flows that represents the User Optimum Dynamic Traffic Assignment (UODTA)

$\Psi(\Xi)$ = vector that represents the path cost resulting from the DTA Ξ

D = set of all feasible flow patterns

3.3 CTM-Related Constraints

The following fundamental CTM constraints constitute the basic traffic dynamics of the subsequent problem formulations of NDP, off-line DTA capacity calibration, and dynamic congestion pricing:

$$x_i^t - x_i^{t-1} + \sum_{(i,j)\in FS(i)} y_{ij}^{t-1} - \sum_{(j,i)\in RS(i)} y_{ji}^{t-1} = d_i^t \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{0,t} \qquad (3.1)$$

$$\sum_{(i,j)\in FS(i)} y_{ij}^t - x_i^t \le 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{1,t} \qquad (3.2)$$

$$\sum_{(j,i)\in RS(i)} y_{ji}^t \le \delta_i^t (N_i^t - x_i^t) \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{2,t} \qquad (3.3)$$

$$\sum_{(j,i)\in RS(i)} y_{ji}^t \le Q_i^t \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{3,t} \qquad (3.4)$$

$$\sum_{(i,j)\in FS(i)} y_{ij}^t \le Q_i^t \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{4,t} \qquad (3.5)$$

$$x_i^0 = \zeta_i \qquad \forall i \in C \setminus C_S \qquad (3.6)$$

$$y_{ij}^0 = 0 \qquad \forall (i,j) \in E \qquad (3.7)$$

$$x_i^{|T|} = 0 \qquad \forall i \in C \setminus C_S \qquad (3.8)$$

$$x_i^t \ge 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad (3.9)$$

$$y_{ij}^t \ge 0 \qquad \forall (i,j) \in E, t \in T \qquad (3.10)$$

Two basic traffic flow relationships are embedded in the above CTM-related constraints. The first is cell mass conservation (Eq. (3.1)), which conserves the flow between cells. The second states that the traffic flow between two cells cannot exceed the number of vehicles occupying the upstream cell (Eq. (3.2)), the remaining capacity of the downstream cell (Eq. (3.3)), and the maximum flow that can exit the upstream cell and enter the downstream cell (Eqs. (3.4)-(3.5)). Equation (3.6) specifies the initial number of vehicles in a CTM network (typically assumed to be zero), while Eq. (3.7) gives the initial flows between cells (typically assumed to be zero as well). Equation (3.8) ensures that all vehicles reach their destinations by the final time interval. The non-negativity constraints are found in Eqs. (3.9)-(3.10). Note that $\pi_i^{0,t}$ - $\pi_i^{4,t}$ represent the dual variables of the corresponding constraints when an appropriate objective function is introduced.

System-Optimal DTA (SODTA) Formulation

Based on the above constraints, the following SODTA formulation was proposed by Ziliaskopoulos (2000) and refined by Waller (2000). For readability and clarity, we present the complete formulation, with similar CTM-related constraints to those previously mentioned.

$$\underset{x,y}{Min} \sum_{(i,j)\in E_S} \sum_{t\in T} \left(t \cdot y_{ij}^t\right) \tag{3.11}$$

subject to

$$x_i^t - x_i^{t-1} + \sum_{(i,j)\in FS(i)} y_{ij}^{t-1} - \sum_{(j,i)\in RS(i)} y_{ji}^{t-1} = d_i^t \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{0,t} \tag{3.12}$$

$$\sum_{(i,j)\in FS(i)} y_{ij}^t - x_i^t \le 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{1,t} \qquad (3.13)$$

$$\sum_{(j,i)\in RS(i)} y_{ji}^t \le \delta_i^t (N_i^t - x_i^t) \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{2,t} \qquad (3.14)$$

$$\sum_{(j,i)\in RS(i)} y_{ji}^t \le Q_i^t \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{3,t} \qquad (3.15)$$

$$\sum_{(i,j)\in FS(i)} y_{ij}^t \le Q_i^t \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{4,t} \qquad (3.16)$$

$$x_i^0 = \zeta_i \qquad \forall i \in C \setminus C_S \qquad (3.17)$$

$$y_{ij}^0 = 0 \qquad \forall (i, j) \in E \qquad (3.18)$$

$$x_i^{|T|} = 0 \qquad \forall i \in C \setminus C_S \qquad (3.19)$$

$$x_i^t \ge 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad (3.20)$$

$$y_{ij}^t \ge 0 \qquad \forall (i, j) \in E, t \in T \qquad (3.21)$$

The objective function of the SODTA is the minimization of the Total System Travel Time (TSTT), which is the difference between the arrival and departure times for every unit of flow within the network. Let γ_i^t be the demand at cell *i* during time interval *t*. Given the departure-time based assumption considered in this single-destination model, the departure time of each demand is known and fixed. Thus, the departure time is a constant equal to $\sum_{i\in C_R}\sum_{t\in T}\left(t\cdot\gamma_i^t\right)$. On the other hand, the arrival time is equal to $\sum_{(i,j)\in E_S}\sum_{t\in T}\left(t\cdot y_{ij}^t\right)$, since a single-destination CTM network is considered in the current formulation. Therefore, the TSTT is $\sum_{(i,j)\in E_S}\sum_{t\in T}\left(t\cdot y_{ij}^t\right)-\sum_{i\in C_R}\sum_{t\in T}\left(t\cdot\gamma_i^t\right)$.

However, as $\sum_{i \in C_R} \sum_{t \in T} \left(t \cdot \gamma_i^t \right)$ is a constant, the upper-level objective function that minimizes the TSTT can be simplified to $\underset{x,y}{Min} \sum_{(i,j) \in E_S} \sum_{t \in T} \left(t \cdot y_{ij}^t \right)$, as shown in Eq. (3.11). Following the CTM theoretical framework, the traffic dynamics can be characterized by Eqs. (3.12)-(3.21).

Ukkusuri (2002) proposed a different objective function to capture the user-optimal behaviors of travelers.

User-Optimal DTA (UODTA) Formulation

$$\underset{x,y}{Min} \sum_{(i,j) \in E_S} \sum_{t \in T} \left(M_t \cdot y_{ij}^t \right) \tag{3.22}$$

subject to

$$x_i^t - x_i^{t-1} + \sum_{(i,j) \in FS(i)} y_{ij}^{t-1} - \sum_{(j,i) \in RS(i)} y_{ji}^{t-1} = d_i^t \qquad \forall i \in C \setminus C_S, t \in T \tag{3.23}$$

$$\sum_{(i,j) \in FS(i)} y_{ij}^t - x_i^t \le 0 \qquad \forall i \in C \setminus C_S, t \in T \tag{3.24}$$

$$\sum_{(j,i) \in RS(i)} y_{ji}^t \le \delta_i^t (N_i^t - x_i^t) \qquad \forall i \in C \setminus C_S, t \in T \tag{3.25}$$

$$\sum_{(j,i) \in RS(i)} y_{ji}^t \le Q_i^t \qquad \forall i \in C \setminus C_S, t \in T \tag{3.26}$$

$$\sum_{(i,j) \in FS(i)} y_{ij}^t \le Q_i^t \qquad \forall i \in C \setminus C_S, t \in T \tag{3.27}$$

$$x_i^0 = \zeta_i \qquad \forall i \in C \setminus C_S \tag{3.28}$$

$$y_{ij}^0 = 0 \qquad \forall (i,j) \in E \tag{3.29}$$

$$x_i^{|T|} = 0 \qquad \forall i \in C \setminus C_S \tag{3.30}$$

$$x_i^t \ge 0 \qquad \forall i \in C \setminus C_S, t \in T \tag{3.31}$$

$$y_{ij}^t \ge 0 \qquad \forall (i,j) \in E, t \in T \tag{3.32}$$

Let γ be the total demand. The vector M_t employed must satisfy the inequality $M_t - M_{t-1} > (M_{|T|} - M_t)\gamma$ to guarantee UO behavior (Ukkusuri, 2002). It is worth noting that M_t is established such that the objective function rewards an individual's earlier arrivals to the extent that it sacrifices overall system costs. One of the deficiencies of the lower-level objective function is that the cost vector M_t grows exponentially, a fact that limits the practical use of such a vector in large-scale implementation. For details regarding the derivation and explanation of M_t, we refer to Ukkusuri (2002).

Traffic management measures are naturally connected with various bi-level programs, in which transportation planners design policies to optimize system performance in the upper-level, while road users attempt to benefit themselves by routing selfishly in the lower-level. In other words, Eq. (3.11) and traffic-related policies/constraints in the upper-level program represent transportation planners' goals and decisions, and thus play the role of leaders in the bi-level program. Eqs. (3.22)-(3.32) characterize road users' behavior, and play the role of followers in a bi-level program. The bi-level program of this form can be viewed as a special case of the Stackelberg game (Von Stackelberg, 1952), in which both leaders and followers can make only one move.

Though a bi-level program can naturally characterize different traffic management measures, several technical challenges arise if one attempts to solve a bi-level program using traditional mathematical programming algorithms due to its non-convex induced region (or feasible set of the upper-level program) and the potential existence of multiple optima. Hence, in this study, we attempt to devise a variety of solution heuristics for tackling bi-level programs, based on dual variable approximations.

For comparison, different meta-heuristics will also be explored to obtain global optimal solutions.

3.4 Re-simulation Dual Approximation Techniques

To the best of our knowledge, there exists no technique to obtain the exact dual variables for the specific forms of the bi-level programs considered in this dissertation. Thus, we adopt the following strategies. Firstly, we solve the UODTA, which is essentially the lower-level program of the bi-level programs (Eqs. (3.22)-(3.32)), and obtain the lower-level solutions (x_i^t, y_{ij}^t). Using the solutions (x_i^t, y_{ij}^t), we approximate the dual variables from the upper-level system-optimal objective function while ignoring the lower-level user-optimal objective function. In other words, the dual variables are approximated via the single-level formulation (Eqs. (3.11)-(3.21)), while the solutions employed (x_i^t, y_{ij}^t) to facilitate the approximations are obtained from the actual lower-level program.

The dual variable $\pi_i^{0,t}$ is defined as the unit change of the objective function value (total system travel time) due to a unit change in the right-hand side of Eq. (3.12), which is the time-dependent demand d_i^t. The dual variable $\pi_i^{0,t}$ can be determined by its problem-specific interpretation in the following manner. First of all, we obtain network flows (x_i^t, y_{ij}^t) of the existing demands, via the UODTA combinatorial algorithm (Waller and Ziliaskopoulos, 2006). The algorithm initially finds the Time-Dependent Shortest Path (TDSP) for all origins to the single destination. The TDSP with the earliest arrival time is then chosen and is assigned to a vehicle. Capacities along that TDSP are reduced accordingly. The process repeats until all demands have been assigned. It is

worth noting that the UODTA combinatorial algorithm employed here has been shown to be able to solve UODTA with reasonable network sizes.

Then, the time-expanded network of the original network is considered, and user paths are simulated on the network to obtain τ_o, the total system travel time of the original time-expanded network. As mentioned earlier, the dual variable $\pi_i^{0,t}$ can be interpreted as the change in the objective function value when the demand associated with cell *i* during time interval *t* is increased by one unit. This is equivalent to the change in total system travel time caused by adding a vehicle to cell *i* during time interval *t*. This additional vehicle follows its TDSP to the destination. Let τ_n denote the total system travel time of the network given the additional demand, which can be determined by simulating the traffic with the additional demand. We then can calculate $\pi_i^{0,t} = \tau_n - \tau_o$. By repeating this process for all cells and time intervals, we obtain $\pi_i^{0,t}$ for all cells and time intervals.

The dual variables $\pi_i^{2,t}$, $\pi_i^{3,t}$ and $\pi_i^{4,t}$ can be determined by a re-simulation process similar to that mentioned above. For instance, the dual variable $\pi_i^{2,t}$ can be interpreted as the change of the objective function value (τ) when the jam density (N_i^t) of cell *i* at time *t* is increased by one unit. We can increase N_i^t by one unit and simulate to calculate τ_n. $\pi_i^{2,t}$ is then equal to $\tau_n - \tau_o$. By repeating this process for all cells and time intervals, we can obtain $\pi_i^{3,t}$ and $\pi_i^{4,t}$ in this manner as well. However, for $\pi_i^{1,t}$, the above process does not apply, since there is no tangible meaning of $\pi_i^{1,t}$. Additionally, the re-simulation process is impractical, especially when large-scale

problems are of interest. Thus, improvements to the dual variable approximation techniques are necessary.

In addition, although these procedures can be used to approximate the dual variables, they are extremely computationally expensive and impractical, as we have to re-simulate for all cells and time intervals to obtain the resulting dual variables. To facilitate dual variable solution strategies, a more efficient dual variable approximation scheme needs to be developed, which will be explored in later sections.

3.5 Summary

This chapter reviews the cell transmission model employed in this dissertation to construct the DTA-based optimization programs. Two basic DTA optimization models, namely SODTA and UODTA, are presented in this chapter to form the bi-level dynamic traffic management problems in later section. Finally, more accurate but impractical re-simulation dual variable approximation techniques are proposed in this chapter. We will improve the dual variable approximation techniques based on the concept investigated in this chapter. The next chapter studies the single-destination Bi-level Linear Programming Network Design Problem (BLPNDP) and designs an efficient and effective Dantzig-Wolfe decomposition based heuristic scheme to tackle BLPNDP.

Chapter 4. Single-destination Bi-level Linear Programming Network Design Problem: A Dantzig-Wolfe Decomposition based Heuristic Scheme

The objective of the transportation network design problem is to determine the optimal capacity expansion policy for a network, subject to a budget constraint. The network design problem is commonly formulated as a bi-level mathematical program resembling a Stackelberg game. In the upper-level, the transportation planners decide on a capacity expansion policy for various network links subject to a budget constraint. In the lower-level, road users react to the capacity changes by selfishly choosing routes that achieve user equilibrium in the modified network. The details of the problem's formulation are as follows.

Bi-level Linear Programming Network Design Problem (BLPNDP) Formulation

$$\underset{x,y}{Min} \sum_{(i,j)\in E_S} \sum_{t\in T} \left(t \cdot y_{ij}^{t}\right) \tag{4.1}$$

subject to

$$\sum_{i\in C\setminus C_S} b_i \le TAB \tag{4.2}$$

$$b_i \ge 0 \qquad \forall i \in C \setminus C_S \tag{4.3}$$

$$\underset{x,y}{Min} \sum_{(i,j)\in E_S} \sum_{t\in T} \left(M_t \cdot y_{ij}^{t}\right) \tag{4.4}$$

$$x_i^t - x_i^{t-1} + \sum_{(i,j)\in FS(i)} y_{ij}^{t-1} - \sum_{(j,i)\in RS(i)} y_{ji}^{t-1} = d_i^t \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{0,t} \tag{4.5}$$

$$\sum_{(i,j)\in FS(i)} y_{ij}^{t} - x_i^t \le 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{1,t} \tag{4.6}$$

$$\sum_{(j,i)\in RS(i)} y_{ji}^t \le \delta_i^t (N_i^t + \chi_i \cdot b_i - x_i^t) \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{2,t} \qquad (4.7)$$

$$\sum_{(j,i)\in RS(i)} y_{ji}^t \le Q_i^t + \phi_i \cdot b_i \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{3,t} \qquad (4.8)$$

$$\sum_{(i,j)\in FS(i)} y_{ij}^t \le Q_i^t + \phi_i \cdot b_i \qquad \forall i \in C \setminus C_S, t \in T \qquad \pi_i^{4,t} \qquad (4.9)$$

$$x_i^0 = \zeta_i \qquad \forall i \in C \setminus C_S \qquad (4.10)$$

$$y_{ij}^0 = 0 \qquad \forall (i,j) \in E \qquad (4.11)$$

$$x_i^{|T|} = 0 \qquad \forall i \in C \setminus C_S \qquad (4.12)$$

$$x_i^t \ge 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad (4.13)$$

$$y_{ij}^t \ge 0 \qquad \forall (i,j) \in E, t \in T \qquad (4.14)$$

In this bi-level program, the leader's objective function (Eq. 4.1) minimizes TSTT, subject to a budget constraint (Eq. 4.2). Other constraints (Eqs. (4.4)-(4.14)) capture the UODTA conditions when the budget is allocated to improve cell capacity. However, instead of using a fixed jam density (N_i^t) and saturation flow rate (Q_i^t), the formulation also needs to incorporate network expansion policies $b_i, \forall i \in C \setminus C_S$ that alter network capacities. As a result (reflected in Eqs. (4.7)-(4.9)), if transportation planners allocate one unit of budget to a cell in a given road segment, the N_i^t and Q_i^t of that particular cell will be increased by χ_i and ϕ_i respectively.

As the underlying structure of the BLPNDP is the UODTA, it may be decomposed. However, there is no known decomposition technique that can be applied directly to this specific bi-level program. We approach the solution by decomposing the

SONDP, which is the single-level LP (Eqs.4.1-4.3 and 4.5-4.14) of BLPNDP without a lower-level objective function (Eq. 4.4). The Dantzig-Wolfe decomposition principle is then applied to the dual formulation of the SONDP such that the resulting master is an LP, and the resulting pricing problem is the dual formulation of the modified SODTA. To reflect the correct user behavior in the BLPNDP, the SODTA pricing problem is replaced by the UODTA. It should be noted that the replacement is a desirable one, since traffic should follow UODTA conditions at the lower level.

We first examine the primal and dual formulations of SONDP which will be used to approximate dual variables in the next subsection. The subsequent subsection discusses the development of the heuristic.

4.1 Primal and Dual Formulations of SONDP

The primal formulation of SONDP is given below:

Primal SONDP

$$\underset{b,x,y}{Min} \quad \sum_{\forall t \in T} \sum_{\forall (i,j) \in E_s} t \cdot y_{ij}^{t} \tag{4.15}$$

subject to

$$\sum_{i \in C \setminus C_S} b_i \le TAB \qquad : \rho \tag{4.16}$$

$$b_i \ge 0 \qquad \forall i \in C \setminus C_S \tag{4.17}$$

$$x_i^t - x_i^{t-1} + \sum_{(i,j) \in FS(i)} y_{ij}^{t-1} - \sum_{(j,i) \in RS(i)} y_{ji}^{t-1} = d_i^t \qquad \forall i \in C \setminus C_S, t \in T \quad : \pi_i^{0,t} \tag{4.18}$$

$$\sum_{(i,j)\in FS(i)} y_{ij}^t - x_i^t \le 0 \qquad \forall i \in C \setminus C_S, t \in T \quad : \pi_i^{1,t} \qquad (4.19)$$

$$\sum_{(j,i)\in RS(i)} y_{ji}^t \le \delta_i^t (N_i^t + \chi_i \cdot b_i - x_i^t) \qquad \forall i \in C \setminus C_S, t \in T \quad : \pi_i^{2,t} \qquad (4.20)$$

$$\sum_{(j,i)\in RS(i)} y_{ji}^t \le Q_i^t + \phi_i \cdot b_i \qquad \forall i \in C \setminus C_S, t \in T \quad : \pi_i^{3,t} \qquad (4.21)$$

$$\sum_{(i,j)\in FS(i)} y_{ij}^t \le Q_i^t + \phi_i \cdot b_i \qquad \forall i \in C \setminus C_S, t \in T \quad : \pi_i^{4,t} \qquad (4.22)$$

$$x_i^0 = \zeta_i \qquad \forall i \in C \setminus C_S \qquad (4.23)$$

$$y_{ij}^0 = 0 \qquad \forall (i, j) \in E \qquad (4.24)$$

$$x_i^{|T|} = 0 \qquad \forall i \in C \setminus C_S \qquad (4.25)$$

$$x_i^t \ge 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad (4.26)$$

$$y_{ij}^t \ge 0 \qquad \forall (i, j) \in E, t \in T \qquad (4.27)$$

The formulation is the BLPNDP without the lower-level objective (Eq.(4.4)). $\pi_i^{0,t}$ - $\pi_i^{4,t}$ are dual variables of the respective constraints Eqs.(4.18)-(4.22); ρ is the dual variable of Eq. (4.16). One of the interesting characteristics of the SONDP formulation is that it is simply the SODTA (Ziliaskopoulos, 2000) with additional columns for budget

variables b_i. By constructing the dual formulation of the primal SONDP, there will be only one constraint (complicating constraint) associated with b_i in the dual formulation. The dual formulation of SONDP is shown below:

Dual SONDP

$$\underset{\pi,\rho}{Max} \sum_{\forall t\in T}\left(\sum_{\forall i\in C_R} d_i^t \pi_i^{0,t} - \sum_{\forall i\in C\setminus(C_R\cup C_S)} \delta_i^t N_i^t \pi_i^{2,t} - \sum_{\forall i\in C\setminus(C_R\cup C_S)} Q_i^t \pi_i^{3,t} - \sum_{\forall i\in C\setminus C_S} Q_i^t \pi_i^{4,t} \right) - \rho \cdot TAB \quad (4.28)$$

subject to

$$\pi_i^{0,t} - \pi_i^{0,t+1} + \pi_i^{1,t} \le 0 \qquad \forall i \in C_R, t \in T \setminus \{|T|\} \qquad : x_i^t \quad (4.29)$$

$$\pi_i^{0,t} - \pi_i^{0,t+1} + \pi_i^{1,t} - \delta_i^t \pi_i^{2,t} \le 0 \qquad \forall i \in C \setminus C_S, t \in T \setminus \{|T|\} \qquad : x_i^t \quad (4.30)$$

$$\pi_i^{0,t+1} - \pi_j^{0,t+1} - \pi_i^{1,t} - \pi_j^{2,t} - \pi_j^{3,t} - \pi_i^{4,t} \le 0 \qquad \forall (i,j) \in E \setminus E_S, t \in T \setminus \{|T|\} \quad : y_{ij}^t \quad (4.31)$$

$$\pi_i^{0,t+1} - \pi_i^{1,t} - \pi_i^{4,t} \le t \qquad \forall (i,j) \in E_S, t \in T \setminus \{|T|\} \qquad : y_{ij}^t \quad (4.32)$$

$$-\rho + \sum_{t\in T} \delta_i^t \chi_i \pi_i^{2,t} + \sum_{t\in T} \phi_i \pi_i^{3,t} + \sum_{t\in T} \phi_i \pi_i^{4,t} \le 0 \qquad \forall i \in C \setminus C_S \qquad : b_i \quad (4.33)$$

$$\pi_i^{0,t} \quad u.r.s. \qquad \forall i \in C \setminus C_S, t \in T \quad (4.34)$$

$$\pi_i^{1,t}, \pi_i^{4,t} \ge 0 \qquad \forall i \in C \setminus C_S, t \in T \quad (4.35)$$

$$\pi_i^{2,t}, \pi_i^{3,t} \geq 0 \qquad \forall i \in C \setminus C_R \cup C_S, t \in T \qquad (4.36)$$

$$\rho \geq 0 \qquad (4.37)$$

We can then decompose this dual formulation by relaxing the complicating constraint (Eq.4.33) according to the Dantzig-Wolfe decomposition principle. The resulting pricing problem is the dual formulation of the well-known SODTA. However, the lower-level program in the original BLPNDP is in fact UODTA. Hence, to reflect the correct user behaviors, we heuristically replace this pricing problem by the UODTA. It can be solved efficiently and optimally by the UODTA combinatorial algorithm (Waller and Ziliaskopoulos, 2006) to obtain the primal variables x_i^t and y_{ij}^t. The next section describes the development of the Dantzig-Wolfe decomposition-based heuristic procedure.

4.2 Algorithmic Design

We decompose the dual formulation of SONDP (Eqs.4.28-4.37) based on the Dantzig-Wolfe decomposition principle. The resulting restricted master program and pricing problem are shown below.

Dantzig-Wolfe Restricted Master Program

$$\underset{w_v,\rho}{Max} \sum_{v=1}^{V}\left(w_v \cdot \sum_{\forall t \in T}\left(\sum_{\forall i \in C_R} d_i^t \pi_i^{0,t,v} - \sum_{\forall i \in C \setminus C_S} \delta_i^t N_i^t \pi_i^{2,t,v} - \sum_{\forall i \in C \setminus C_S} Q_i^t \pi_i^{3,t,v} - \sum_{\forall i \in C \setminus C_S} Q_i^t \pi_i^{4,t,v} \right) \right) - \rho \cdot TAB$$

(4.38)

subject to

$$-\rho+\sum_{v=1}^{V}\left(w_v\cdot\left(\sum_{t\in T}\delta_i^t\chi_i\pi_i^{2,t,v}+\sum_{t\in T}\phi_i\pi_i^{3,t,v}+\sum_{t\in T}\phi_i\pi_i^{4,t,v}\right)\right)\le 0 \qquad \forall i\in C\setminus C_S \qquad :b_i \qquad (4.39)$$

$$\sum_{v=1}^{V}w_v=1 \qquad :g \qquad (4.40)$$

$$w_v\ge 0 \qquad \forall v=1,2,...,V \qquad (4.41)$$

$$\rho\ge 0 \qquad (4.42)$$

The master problem is a budget allocation problem and can be solved with virtually any LP solver. The dual variables of Eq. (4.39) provide the budget allocation policies ($b_i \ \forall i\in C\setminus C_S$) of the current iteration. Then the budget allocation is passed to the pricing problem. To be specific, in our numerical experiments, we solve the restricted master problem with the CPLEX solver and read the dual variables of Eq. (4.39) as b_i in each iteration. The dual variables b_i are then passed to the pricing problem, and change capacities in the pricing problem accordingly. The Dantzig-Wolfe decomposition pricing problem is shown below:

Dantzig-Wolfe Pricing Problem

$$\underset{\pi_i^{0,t},\pi_i^{1,t},\pi_i^{2,t},\pi_i^{3,t},\pi_i^{4,t}}{Max}\ \sum_{\forall t\in T}\left(\sum_{\forall i\in C_R}d_i^t\pi_i^{0,t}-\sum_{\forall i\in C\setminus C_S}\delta_i^tN_i^t\pi_i^{2,t}-\sum_{\forall i\in C\setminus C_S}Q_i^t\pi_i^{3,t}-\sum_{\forall i\in C\setminus C_S}Q_i^t\pi_i^{4,t}\right)$$

$$- \sum_{i \in C \setminus C_S} \left(b_i \cdot \left(\sum_{t \in T} \delta_i^t \chi_i \pi_i^{2,t} + \sum_{t \in T} \phi_i \pi_i^{3,t} + \sum_{t \in T} \phi_i \pi_i^{4,t} \right) \right) - g \qquad (4.43)$$

subject to

$$\pi_i^{0,t} - \pi_i^{0,t+1} + \pi_i^{1,t} \leq 0 \qquad \forall i \in C_R, t \in T \setminus \{|T|\} \qquad : x_i^t \quad (4.44)$$

$$\pi_i^{0,t} - \pi_i^{0,t+1} + \pi_i^{1,t} - \delta_i^t \pi_i^{2,t} \leq 0 \qquad \forall i \in C \setminus C_S, t \in T \setminus \{|T|\} \qquad : x_i^t \quad (4.45)$$

$$\pi_i^{0,t+1} - \pi_j^{0,t+1} - \pi_i^{1,t} - \pi_j^{2,t} - \pi_j^{3,t} - \pi_i^{4,t} \leq 0 \qquad \forall (i,j) \in E \setminus E_S, t \in T \setminus \{|T|\} \quad : y_{ij}^t \quad (4.46)$$

$$\pi_i^{0,t+1} - \pi_i^{1,t} - \pi_i^{4,t} \leq t \qquad \forall (i,j) \in E_S, t \in T \setminus \{|T|\} \qquad : y_{ij}^t \quad (4.47)$$

$$\pi_i^{0,t} \text{ u.r.s.} \qquad \forall i \in C \setminus C_S, t \in T \qquad (4.48)$$

$$\pi_i^{1,t}, \pi_i^{4,t} \geq 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad (4.49)$$

$$\pi_i^{2,t}, \pi_i^{3,t} \geq 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad (4.50)$$

The pricing problem is a SODTA linear program that was modified by replacing the original Q_i^t and N_i^t with $Q_i^t + \phi_i b_i$ and $N_i^t + \chi_i b_i$, respectively. This problem is replaced by the modified UODTA as the traveler behavior of BLPNDP is actually UODTA. With the budget allocation policies passed from the master program, the resulting pricing problems can be solved efficiently by the UODTA combinatorial

algorithm. For the details of the UODTA combinatorial algorithm, we refer the reader to Waller and Ziliaskopoulos (2006). The objective function (Eq.4.43) of the pricing problem also serves as the stopping criterion of the Dantzig-Wolfe decomposition based heuristic scheme. The complete heuristic scheme is outlined below:

Data: $N_i^{t,oritinal}, Q_i^{t,original}, d_i^t, \delta_i^t \quad \forall i \in C, t \in T$; $\chi_i, \phi_i \forall i \in C \setminus C_S; TAB$

Step 0: Set $v = 1$;

Set $N_i^t = N_i^{t,original} \qquad \forall i \in C, t \in T$

Set $Q_i^t = Q_i^{t,original} \qquad \forall i \in C, t \in T$

Set $b_i = 0 \qquad \forall i \in C \setminus C_S$

Set $g = 0$

Step 1: Solve the UODTA with the combinatorial algorithm with N_i^t and Q_i^t

$\forall i \in C, t \in T$ to obtain $x_i^t \ \forall i \in C \setminus C_S, t \in T$, $y_{ij}^t \ \forall (i, j) \in E, t \in T$.

Step 2: Approximate the dual variables $\pi_i^{0,t,v}, \pi_i^{1,t,v}, \pi_i^{2,t,v}, \pi_i^{3,t,v}, \pi_i^{4,t,v}$ based on the solution in Step 1

Step 3: If the objective function value of the pricing problem is less than or equal to zero, stop. That is, if $\sum_{\forall t \in T}\left(\sum_{\forall i \in C_R} d_i^t \pi_i^{0,t,v} - \sum_{\forall i \in C \setminus C_S} \delta_i^t N_i^t \pi_i^{2,t,v} - \sum_{\forall i \in C \setminus C_S} Q_i^t \pi_i^{3,t,v} - \sum_{\forall i \in C \setminus C_S} Q_i^t \pi_i^{4,t,v} \right)$

$- \sum_{i \in C \setminus C_S} \left(b_i \cdot \left(\sum_{t \in T} \delta_i^t \chi_i \pi_i^{2,t,v} + \sum_{t \in T} \phi_i \pi_i^{3,t,v} + \sum_{t \in T} \phi_i \pi_i^{4,t,v} \right) \right) - g \le 0$, stop and report the solution $b_i \ \forall i \in C \setminus C_S$, $x_i^t \ \forall i \in C \setminus C_S, t \in T$, $y_{ij}^t \ \forall (i, j) \in E, t \in T$. Otherwise,

go to Step 4.

Step 4: Solve the restricted master program with CPLEX to obtain the dual variables of the master program: $b_i \ \forall i \in C \setminus C_S$ and g. Denote the set of cells within the same link of cell i by l_i; and the allocated budget to cell k by $bp_k \ \forall k \in C \setminus C_S$, which are determined from the following procedure:

Set $bp_k = 0 \quad \forall k \in C \setminus C_S$

For all $i \in C \setminus C_S$, if $b_i > 0$, set $bp_k = bp_k + b_i / |l_i| \quad \forall k \in l_i$.

Step 5: Set $v = v + 1$;

Set $N_i^t = N_i^{t,original} + \chi_i \cdot bp_i$;

Set $Q_i^t = Q_i^{t,original} + \phi_i \cdot bp_i$.

Go to Step 1

The heuristic scheme iteratively solves the pricing problem (UODTA) and augments the columns in the restricted master problem using the UODTA solution. The restricted master problem then decides the budget allocation and passes the allocation back to the pricing problem. The allocated budget changes the capacities in the pricing problem, and the process continues. The process is repeated until the stopping criterion is met.

It should be noted that downstream bottlenecks have great impacts on upstream traffic when CTM is employed to capture traffic dynamics. For instance, on a freeway link, a bottleneck can take place in downstream cells when only the upstream cells are

expanded. Therefore, when $b_i > 0$ in step 4, we allocate b_i evenly over the cells within a link to avoid traffic bottleneck due to the theoretical CTM. For the details of this solution strategy, we refer the reader to Karoonsoontawong (2006).

The Dantzig-Wolfe decomposition-based heuristic scheme proposed here relies heavily on dual variables. However, to the best of our knowledge, there is no procedure to obtain the exact dual variables for the BLPNDP. Therefore, the required dual variables ($\pi_i^{0,t}, \pi_i^{1,t}, \pi_i^{2,t}, \pi_i^{3,t}, \pi_i^{4,t}$) are approximated. Details of the improved dual approximation techniques are discussed in the next section.

4.3 Improved Dual Variables Approximation Techniques

In this section, we first describe the approximation of $\pi_i^{0,t}$, followed by the approximations of $\pi_i^{1,t}, \pi_i^{2,t}, \pi_i^{3,t}$, and $\pi_i^{4,t}$. The approximations of $\pi_i^{1,t}, \pi_i^{2,t}, \pi_i^{3,t}$, and $\pi_i^{4,t}$ are developed based on the approximation of $\pi_i^{0,t}$.

4.3.1. Approximation of $\pi_i^{0,t}$

We determine $\pi_i^{0,t}$ using the problem-specific interpretation of this dual variables. With the original network, we first obtain the user paths with the existing demands via the UODTA combinatorial algorithm. The dual variable $\pi_i^{0,t}$ can be interpreted as the change in the objective function value (total system travel time, τ) when the demand of cell *i* at time interval *t* is increased by one unit. This is equivalent to the change in τ caused by adding a vehicle in cell *i* at time interval *t*. This additional vehicle follows the TDSP to the destination. Let τ_n denote the total system travel time of the network with the additional demand. We then can

calculate $\pi_i^{0,t} = \tau_n - \tau_o$. By repeating this process for all cells and time intervals, we can obtain $\pi_i^{0,t}$ for all cells and time intervals.

However, as mentioned before, the re-simulation approximation process above is extremely computationally expensive, and impractical for BLPNDP when examining large-scale networks. One of the main disadvantages of the process is that we have to re-simulate for all cells and time intervals to obtain the resulting dual variables. To address this issue, we approximate $\pi_i^{0,t}$ by setting it equal to the travel time of the additional user's TDSP. Using the travel time of TDSP as $\pi_i^{0,t}$ significantly decreases the computational burden and is a more practical measure. However, finding the TDSP explicitly can also be computationally expensive and is not necessary for the heuristic scheme. Therefore, we propose a backward connectivity algorithm to obtain the travel time of the TDSP without explicitly finding the path itself.

The proposed backward connectivity algorithm is described as follows. In the time-expanded network, the sink cell is assigned an index $(T - t + 1)$ according to its time interval (t), and the labels of all other cells are set to zero. The algorithm checks the connectivity from a sink cell to its upstream cells. Labels are assigned to upstream cells that have smaller labels and can be reached from that sink cell. The algorithm then moves to the recently updated cell and updates its upstream cells in the same manner. The process is implemented in a dequeue data structure (Ahuja, Magnanti, and Orlin, 1993) and repeats recursively until it reaches any source cell. During the searching process, only the latest label is kept in each cell. The travel time of the TDSP from one cell at a particular time interval to its destination is then equal to the difference between the final label and the index of the sink cell at this time interval.

To demonstrate the proposed backward connectivity algorithm, we utilize the time-expanded network in FIGURE 1 as an example. In this example, the planning period T is assumed to be 7, and the number of cells is 6. The source cells in this network are cell 1 and cell 2, while the sink cell is cell 6. The indices of sink cells $(T-t+1)$ are given on the right of the figure. Initially, the sink cell is assigned the label equal to its index at the same time interval. For example, cell 6 is assigned a label of 7 at time 1. The labels of other cells are set to be zero before the algorithm begins. We assume that the bottleneck happens in cell 5 at time 4.

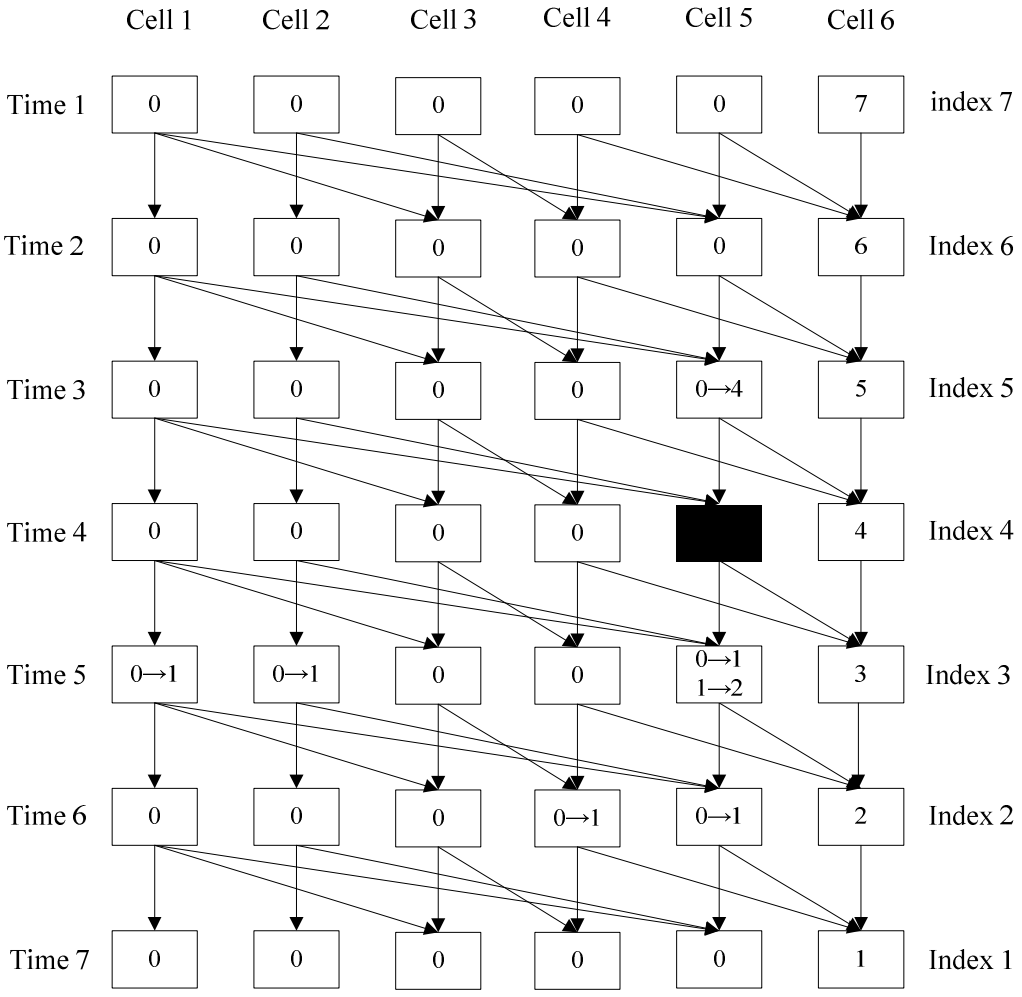

FIGURE 1: 6-cell CTM Time-expanded Network

Suppose that the algorithm is currently searching cell 6 at time 7. The labels of cells 4 and 5 at time 6 can be updated to 1, since these cells can be reached from cell 6 at time 7 and have smaller labels (label 0 < label 1). Cells 4 and 5 are then placed into the scan eligible list and can update their upstream cells in later iterations. For example, cell 5 can later update cell 1, 2, and 5 at time 5 (from 0→1). However, the label of cell 5 at time 5 can be further updated (from 1→2) from cell 6 at time 6, since it has a smaller label (label 1 < label 2). If a bottleneck is met, the upstream cells cannot be updated. For instance, the algorithm cannot update cell 5 at time 4 from cell 6 at time 5; similarly, cell 5 at time 3 can only be updated from cell 6 at time 4, not cell 5 at time 4. The searching process repeats recursively until it reaches any source cells (i.e., either cell 1 or cell 2 in this example). The travel time of the TDSP in cell 5 at time 6, for example, is index (2) – label (1) = 1 if the searching process completes and labels are finalized. The travel time of the TDSP is the dual variable of the constraint corresponding to the cell.

Specifically, we present the pseudo-code. The notations used in the mathematical formulation are also employed here. In addition, $\Gamma^{-1}(j,t)$ denotes the cells upstream of cell *j* at time *t* in the time-expanded network; DL is the destination list; and SE is the scan eligible list. The pseudo-code for the backward connectivity algorithm is given below:

Initialization

Set $Index(j,t) = T - t + 1 \qquad \forall j \in C_S, t \in T$

$Label(j,t) = Index(j,t) \qquad \forall j \in C_S, t \in T$

$Label(i,t) = 0 \qquad \forall i \in C \setminus C_S, t \in T$

Main loop

For all $j \in C_S, t \in T$

Set $DL = \{(j,t) \mid j \in C_S, t \in T\}$

While *DL* is not empty

Remove *(j,t)* from *DL*

Set $t_D = Label(j,t)$

Insert *(j,t)* into *SE*

While *SE* is not empty

Remove *(j,t)* from *SE*

For all $i \in \Gamma^{-1}(j,t), \ i \in C \setminus C_S$

If $N_i^{t-1} - x_i^{t-1} > 0$ and $Q_{ij}^{t-1} - y_{ij}^{t-1} > 0$ and $Label(i,t-1) < t_D$, then

Set $Label(i,t-1) = t_D$

Insert *(i,t-1)* into *SE*

End if

End for

End while

End while

End for

$Dual(i,t) = Index(j,t) - Label(i,t) \quad \forall i \in C \setminus C_S, t \in T, j \in C_S$

Report $Dual(i,t) \quad \forall i \in C \setminus C_S, t \in T$

The algorithm efficiently finds the dual variables $\pi_i^{0,t}$, without re-simulating or explicitly finding the TDSP. In our preliminary experiments, the computational time of the heuristic scheme decreased significantly with this improvement.

4.3.2. Approximation of $\pi_i^{1,t}, \pi_i^{2,t}, \pi_i^{3,t}$ *and* $\pi_i^{4,t}$

The dual variables $\pi_i^{2,t}$, $\pi_i^{3,t}$, and $\pi_i^{4,t}$ can be determined by the same re-simulation process presented in previous chapter. However, for $\pi_i^{1,t}$, the re-simulation process cannot be applied, as there is no tangible meaning of $\pi_i^{1,t}$ and the re-simulation process is once again impractical. Hence, we devise a heuristic process using the complementary slackness properties of Linear Programming (LP) theory.

Based on the complementary slackness conditions, the dual variables $\pi_i^{1,t}$ $\pi_i^{2,t}$, $\pi_i^{3,t}$, and $\pi_i^{4,t}$ are set to zero when the associated constraints are non-binding; otherwise, they are known only to be non-negative. When we consider the primal formulation of SONDP, the following conditions are established from the complementary slackness properties for Eqs.(4.19)-(4.22):

$$\text{If } \sum_{(i,j)\in FS(i)} y_{ij}^t - x_i^t < 0, \text{ then set } \pi_i^{1,t} = 0 \quad \forall i \in C \setminus C_S, t \in T$$

$$\text{If } \sum_{(j,i)\in RS(i)} y_{ji}^t < \delta_i^t (N_i^t + \chi_i \cdot b_i - x_i^t), \text{ set } \pi^{2,t} = 0 \quad \forall i \in C \setminus C_S, t \in T$$

$$\text{If } \sum_{(j,i)\in RS(i)} y_{ji}^t < Q_i^t + \phi_i \cdot b_i, \text{ set } \pi_i^{3,t} = 0 \quad \forall i \in C \setminus C_S, t \in T$$

If $\sum_{(i,j)\in FS(i)} y_{ij}^t < Q_i^t + \phi_i \cdot b_i$, set $\pi_i^{4,t} = 0 \quad \forall i \in C \setminus C_s, t \in T$

Furthermore, when we consider the dual formulation of SONDP, the complementary slackness conditions for (Eq.4.29) are employed to determine $\pi_i^{1,t}$ $\forall i \in C_R, t \in T \setminus \{|T|\}$:

If $x_i^t > 0$, set $\pi_i^{1,t} = \pi_i^{0,t+1} - \pi_i^{0,t}$

Else if $x_i^t = 0$, we approximate $\pi_i^{1,t}$ by the equality $\pi_i^{1,t} = \pi_i^{0,t+1} - \pi_i^{0,t}$

Subsequently, the complementary slackness conditions for (Eq.4.30) are utilized to determine $\pi_i^{1,t}$ and $\pi_i^{2t} \quad \forall i \in C \setminus C_S, t \in T \setminus \{|T|\}$:

If $x_i^t > 0$, set $\pi_i^{1,t} - \delta_i^t \pi_i^{2,t} = \pi_i^{0,t+1} - \pi_i^{0,t}$

Else if $x_i^t = 0$, we approximate the dual variables by treating the inequality $\pi_i^{1,t} - \delta_i^t \pi_i^{2,t} \le \pi_i^{0,t+1} - \pi_i^{0,t}$ as an equality. If $\pi_i^{1,t}$ and π_i^{2t} are both undetermined, we assume $\pi_i^{1,t} = 0$ and $\pi_i^{2t} = -(\pi_i^{0,t+1} - \pi_i^{0,t})$

π_i^{2t} represents the change in τ when the right-hand-side of (Eq.4.20) changes by one unit. Assuming $\pi_i^{2t} = -(\pi_i^{0,t+1} - \pi_i^{0,t})$ is reasonable, since an increase in the

right-hand-side of (Eq.4.20) is equivalent to an increase in the jam density; the increase in the jam density should potentially decrease τ.

Next, the complementary slackness conditions for (Eq.4.31) are employed to determine $\pi_j^{3,t}$ and $\pi_i^{4t} \quad \forall (i, j) \in E \setminus E_S, t \in T \setminus \{|T|\}$:

If $y_{ij}^t > 0$, $\pi_j^{3,t} + \pi_i^{4,t} = \pi_i^{0,t+1} - \pi_j^{0,t+1} - \pi_i^{1,t} - \pi_j^{2,t}$

Else if $y_{ij}^t = 0$, we approximate the dual variables by treating the inequality $\pi_j^{3,t} + \pi_i^{4,t} \geq \pi_i^{0,t+1} - \pi_j^{0,t+1} - \pi_i^{1,t} - \pi_j^{2,t}$ as an equality. If $\pi_j^{3,t}$ and π_i^{4t} are both undetermined, we assume $\pi_j^{3,t} = \pi_i^{4t}$

Finally, the complementary slackness conditions for (Eq.4.32) are employed to determine $\pi_i^{4t} \quad \forall (i, j) \in E_S, t \in T \setminus \{|T|\}$:

If $y_{ij}^t > 0$, $\pi_i^{4,t} = \pi_i^{0,t+1} - \pi_i^{1,t} - t$

Else if $y_{ij}^t = 0$, we approximate the dual variables by treating the inequality $\pi_i^{4,t} \geq \pi_i^{0,t+1} - \pi_i^{1,t} - t$ as an equality

The next section shows the results of numerical experiments on two test networks to demonstrate the applicability and computational efficiency of the proposed Dantzig-Wolfe decomposition-based heuristic scheme.

4.4 Numerical Experiments

The Dantzig-Wolfe decomposition-based heuristic scheme is tested on the 6-cell CTM network illustrated in FIGURE 2, and the 68-cell CTM network in FIGURE 3. In the following experiments, we stop the Dantzig-Wolfe decomposition-based heuristic scheme when either the stopping criterion is met or the iteration limit (10) is reached. The programs of the Dantzig-Wolfe decomposition-based heuristic scheme, i.e., the UODTA combinatorial algorithm and the dual variable approximation techniques are implemented in the standard ANSI C language. The numerical experiments are conducted on a Linux machine with an Intel 3.00GHz Xeon CPU and 32 GB of memory.

The experiments are primarily for demonstrating the applicability of the heuristic scheme. Parameters used in the experiments are not selected to show any specific results and can be modified without affecting the applicability.

4.4.1. 6-cell CTM Network

The 6-cell CTM network (FIGURE 2) is composed of 6 cells and 6 cell connectors. There are two source cells (cell 1 and cell 2) and one sink cell (cell 6) in the network. Except for the sink cell 6, all cells are considered for capacity expansion.

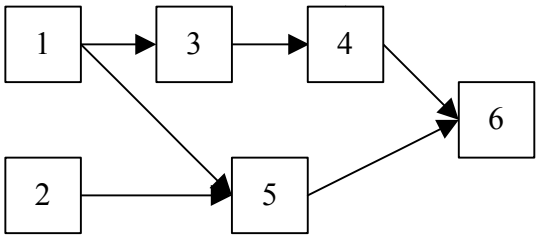

FIGURE 2: 6-cell CTM Network

The time-varying characteristics of the network (jam density N_i^t, saturation flow rate Q_i^t) and network improvement parameters (χ_i , ϕ_i) are described in TABLE 1:

TABLE 1: Characteristics of 6-cell CTM Network

Cell	N_i^t	Q_i^t	χ_i	ϕ_i
1	+inf	1	1	1
2	+inf	1	1	1
3	2	1	1	1
4	2	1	1	1
5	2	1	1	1
6	+inf	+inf	-	-

The planning period is eight time intervals, with two OD pairs. The seven time-dependent OD demands are given in TABLE 2:

TABLE 2: Time-dependent OD Demands for 6-cell CTM Network

	Origin	Destination
Time 1	1	6
	1	6
	1	6
	2	6
Time 2	1	6
	1	6
	2	6

With an iteration limit of ten, the solutions obtained from the Dantzig-Wolfe decomposition-based heuristic scheme with different budget constraints are summarized in TABLE 3. The computational times required for the proposed scheme are less than one second for all cases in this experiment. In TABLE 3, we also present the optimal solutions from the modified K^{th}-Best algorithm (Karoonsoontawong and Waller, 2006).

TABLE 3: 6-cell CTM Network Cell Expansion Policies with Different Budget

TAB (unit)	DWD Based Heuristic Scheme						Optimal K^{th}-Best Algorithm					
	τ	b[1]	b[2]	b[3]	b[4]	b[5]	τ	b[1]	b[2]	b[3]	b[4]	b[5]
50	14.00	20.63	0	0	0	29.37	14.00	2.00	0	0	0	5.00
40	14.00	16.42	0	0	0	23.58	14.00	2.00	0	0	0	5.00
30	14.00	12.21	0	0	0	17.79	14.00	2.00	0	0	0	5.00
20	14.00	8.00	0	0	0	12.00	14.00	2.00	0	0	0	5.00
10	14.00	3.79	0	0	0	6.21	14.00	2.00	0	0	0	5.00
9	14.00	3.37	0	0	0	5.63	14.00	2.00	0	0	0	5.00
8	14.00	2.95	0	0	0	5.05	14.00	2.00	0	0	0	5.00
7	14.36	2.13	0.23	0	0	4.64	14.00	2.00	0	0	0	5.00
6	15.11	2.11	0	0	0	3.89	15.00	2.00	0	0	0	4.00
5	16.01	1.68	0	0	0	3.32	16.00	2.00	0	0	0	3.00
4	17.01	1.26	0	0	0	2.74	17.00	2.00	0	0	0	2.00
3	18.02	0.84	0	0	0	2.16	18.00	1.50	0	0	0	1.50
2	20.02	0.42	0	0	0	1.58	19.00	1.00	0	0	0	1.00
1	23.00	0	0	0	0	1.00	21.00	0.333	0	0	0	0.667

We compare the two approaches based on the total system travel time τ. When an approach obtains lower τ under an identical budget level *TAB*, we assert that this approach outperforms the other.

The Dantzig-Wolfe decomposition-based heuristic scheme obtains the optimal solutions when the budget levels are high ($TAB \geq 8$). However, the modified K^{th}-Best algorithm outperforms the Dantzig-Wolfe decomposition based heuristic scheme when budget levels are low. The reason behind the worse performance of the Dantzig-Wolfe decomposition-based heuristic scheme is likely the use of the approximated dual variables. As mentioned before, to our knowledge, there is no procedure available to obtain the exact dual variables for the bi-level linear programs in an efficient manner. While the results may appear reasonable, further research is still necessary to improve the dual approximation procedure.

Although the modified K^{th}-Best algorithm has better performance in terms of solution quality, it is not suitable for large-scale deployment. The objective function of the lower-level problem (UODTA) and the bi-level nature of the overall problem limit the problem size that can be solved by the modified K^{th}-Best algorithm. Both characteristics cause the difficulty of solving this problem grows exponentially. Thus, it is not possible to guarantee an optimal solution to larger problems with this method. For continued comparison of the Dantzig-Wolfe decomposition heuristic we, therefore, employ a meta-heuristic algorithm to obtain comparable solutions when the network size is larger.

The Dantzig-Wolfe decomposition-based heuristic scheme can solve larger BLPNDP than the modified K^{th}-Best algorithm for three main reasons. First, we adopted a combinatorial algorithm in solving UODTA. The algorithm is proven to be able to optimally solve UODTA with reasonable size networks. Secondly, the dual approximation procedures are efficient since we employ a backward connectivity algorithm and complementary slackness properties. The procedures can approximate the dual variables efficiently, even when large-scale networks are under consideration. Finally, the proposed scheme is relatively flexible. For instance, instead of finding dual variables for one single cell, we can approximate them for an entire segment of highway and save a considerable amount of computational time. The flexibility enables us to solve even larger problems.

A trend can be seen that the budget is not completely used in the K^{th}-Best algorithm solution at certain budget levels, while in the heuristic scheme the entire budget is used. The design of the heuristic scheme and the K^{th}-Best algorithm is behind this difference. The Dantzig-Wolfe decomposition-based heuristic scheme is designed to employ the entire budget to get the lowest τ. On the other hand, the K^{th}-Best

algorithm is designed to obtain the lowest τ, with the sum of allocated budgets being constrained to less than or equal to the total available budget. This observation indicates one potential direction for future improvement.

4.4.2. 68-cell CTM Network

The 68-cell CTM network (FIGURE 3) is composed of 68 cells and 74 cell connectors. There are three source cells (cells 1, 2, and 3) and one sink cell (cell 68). Except for the sink cell 68, all cells are considered for capacity expansion. The budget level TAB for this network is 100 units.

The cells in the center represent the freeway, and the outer and cross cells represent arterial streets. The deterministic OD demand and time-varying network characteristics of the network are described in TABLE 4 and TABLE 5 respectively.

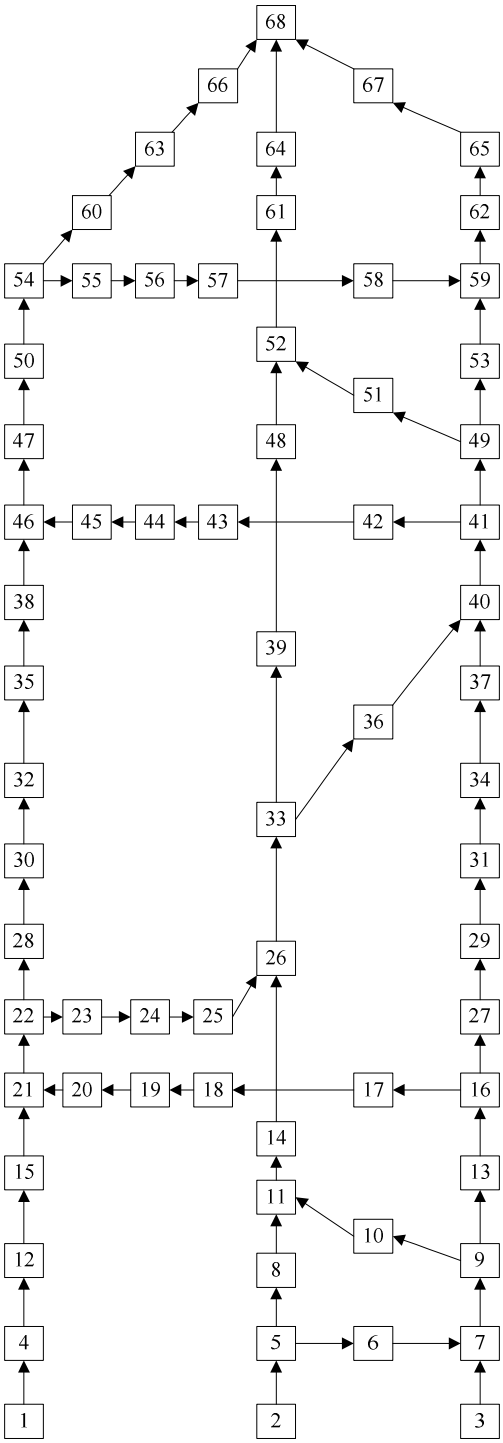

FIGURE 3: 68-cell CTM Network

TABLE 4: Time-dependent OD Demand for 68-cell CTM Network

Cell	Destination	Demand (vph[1])
1	68	1,800
2	68	3,600
3	68	1,800

[1] demand is uniformly distributed over the planning period

TABLE 5: Characteristics of 68-cell CTM Network

Cell	N_i^t	Q_i^t	χ_i	ϕ_i
1, 3	+inf	8	1	1
2	+inf	12	1	1
68	+inf	+inf	-	-
Freeway Cells[1]	20	12	1	1
Arterial Cells[2]	10	8	1	1

[1] other than cell 2 and cell 68
[2] other than cell 1 and cell 3

In the Dantzig-Wolfe decomposition-based heuristic scheme, each iteration includes one functional evaluation (UODTA). The heuristic scheme is terminated if the stopping criterion is met or the iteration limit is reached. Using these parameters and a *TAB* of 100, the Dantzig-Wolfe decomposition-based heuristic scheme obtains a solution with total system travel time τ = 6,639 within a CPU time of 588.524 seconds. We attempted to improve the solution by increasing the iteration limit, but τ could not be further improved after 10 iterations.

To obtain a solution comparable to the Dantzig-Wolfe decomposition (DWD) based heuristic scheme, we employed the Genetic Algorithm (GA) used in Karoonsoontawong and Waller (2006) to solve this problem. The GA parameters include population size (50), crossover rate (0.6), and mutation rate (0.001). These are the calibrated parameters in Karoonsoontawong and Waller (2006). In the GA, τ is used as the fitness measure in each functional evaluation. The number of functional evaluations and the incumbent τ of both methods are presented in FIGURE 4.

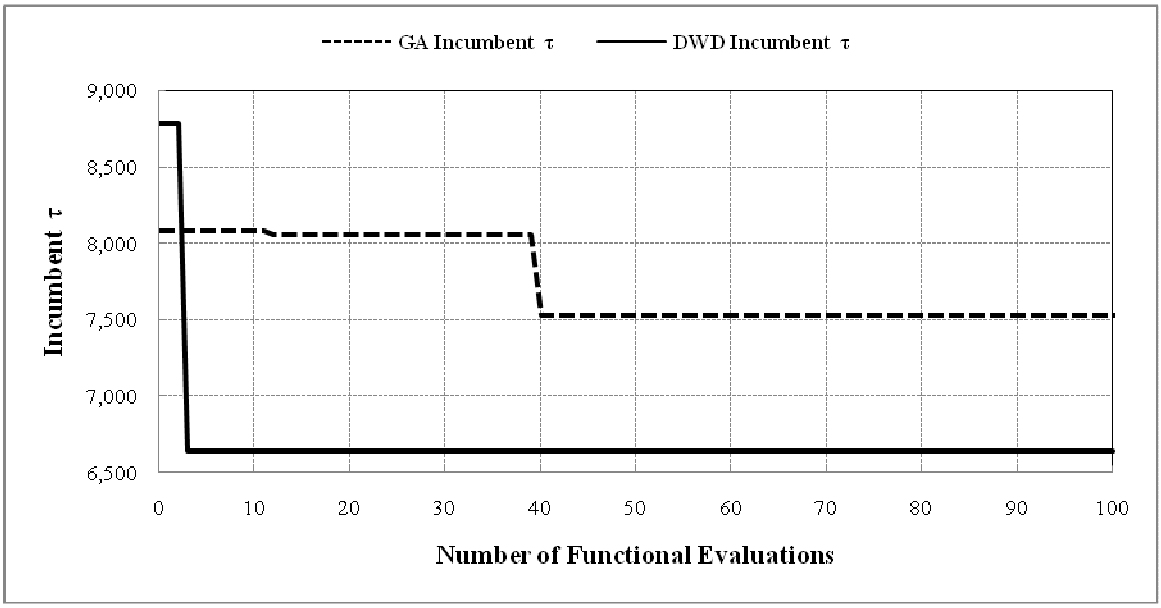

FIGURE 4: Comparison of Algorithms Performance ($TAB = 100$)

From FIGURE 4, we can see that the Dantzig-Wolfe decomposition-based heuristic scheme starts with higher τ in the beginning and quickly finds its best solution in iteration 3. The solution found in iteration 3 is either a local optimum or possibly a globally optimal solution. Contrary to this, within 100 function evaluations, the GA solutions improve slowly, and it has achieved its best solution at functional evaluation 40 (τ = 7,530). As can be seen, the solution quality of the GA is worse than that of the Dantzig-Wolfe decomposition-based heuristic scheme. Furthermore, FIGURE 4 shows that it takes more functional evaluations for GA to converge to an equivalent solution; more functional evaluations imply more computational time.

To further demonstrate the computational effort required for GA, we increase the number of functional evaluations allowed and summarize the results in TABLE 6. With 10,000 function evaluations, τ still improves slowly in the GA approach. Finally, the GA solution method finds its best solution (τ = 6,639) with 9,752 function evaluations in 135.281 hours, while the Dantzig-Wolfe decomposition based heuristic scheme finds

the equivalent solution (τ = 6,639) with 10 function evaluations in 9.809 minutes. The computational effort required impedes the practical use of GA, especially when large-scale BLPNDPs are of interest.

TABLE 6: Computational Time for GA

No. of Function Evaluations	Incumbent τ	CPU Time (seconds)	No. of Function Evaluations	Incumbent τ	CPU Time (seconds)
1	8,082	49	4,177	7,468	202,893
10	8,080	485	4,302	7,467	208,964
12	8,056	582	4,360	7,034	211,780
40	7,530	1,941	5,188	7,032	251,997
390	7,521	18,918	5,633	7,027	273,603
504	7,511	24,448	6,671	7,025	324,010
610	7,502	29,588	6,945	7,024	337,265
1,526	7,499	74,049	8,325	7,021	404,587
1,658	7,489	80,469	9,417	7,018	458,223
2,023	7,482	98,187	9,641	7,016	469,418
2,925	7,477	142,021	9,752	6,639	474,923
3,962	7,472	192,452	10,000	6,639	487,012

However, it can be verified that the τ of 6,639 is the free flow total system travel time; which implies that the budget level of 100 is relatively high. To further test the stringent budgetary constraint, we reduce the budget to 50 and run both the Dantzig-Wolfe decomposition based heuristic scheme and the GA approach with 100 functional evaluations. The results are depicted in FIGURE 5.

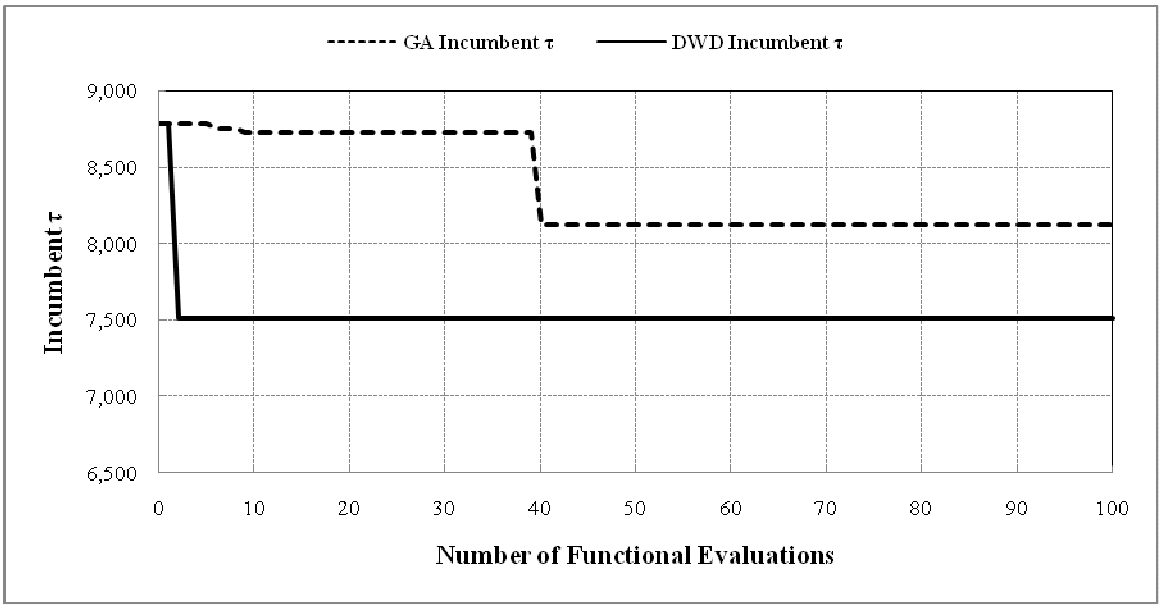

FIGURE 5: Comparison of Algorithms Performance ($TAB = 50$)

As can be seen from FIGURE 5, the Dantzig-Wolfe decomposition based heuristic scheme still outperforms the GA approach even with the more stringent budgetary constraint. In addition, the Dantzig-Wolfe decomposition based heuristic scheme still converges to a better solution with fewer functional evaluations.

From the results above, we can see that the Dantzig-Wolfe decomposition based heuristic scheme outperforms the GA approach although the heuristic scheme possibly stays in a local optimum. Further, the computational effort required by the Dantzig-Wolfe decomposition based heuristic scheme is reasonable when compared to an optimal approach or the GA solution method. The results reveal the promising applicability of the heuristic scheme to larger network.

4.5 Summary

The BLPNDP is bi-level by nature and is known to be NP-complete. We proposed the Dantzig-Wolfe decomposition based heuristic scheme in this chapter to solve this problem. The heuristic scheme can address the computational issues that limit

the practical use of existing approaches. Potentially, the scheme can solve large-scale BLPNDP because of the flexible design and efficient dual approximation techniques.

Although the BLPNDP can be solved by the proposed heuristic scheme, it is possible to further improve the solution quality. The difference between solutions from the heuristic scheme and the exact solution may be due to the use of approximated dual variables instead of exact dual variables. In addition, by observing the solution behavior, we can see that the solutions found by proposed heuristic method can be potentially local optimum. Further investigation is necessary to minimize the deviations in order to improve the solution quality.

Moreover, the scope of this chapter is limited to multiple-origin and single-destination BLPNDP. The intent is to investigate the premise that the heuristic scheme proposed can provide a good solution within reasonable computational time. The natural extension of this study would be multiple-origin and multiple-destination BLPNDP.

Lastly, the dual approximation procedures developed in this chapter is not of exclusive use of the BLPNDP. Potentially, these procedures can be used in fields such as roadway congestion pricing. It is a widely used concept that the price of roadway is set based on marginal social cost. The marginal social cost can be approximated through the same dual approximation procedure. Further applications of this procedure will be explored in later sections.

Chapter 5. Single-destination Bi-level Linear Programming Network Design Problem: A Dual Approximation Genetic Algorithm

In this section, we detail the design of the Dual Approximation Genetic Algorithm (DAGA) for solving the BLPNDP. Genetic Algorithm (GA) is a global search heuristic developed on the principle of evolutionary biology, in which regions having a higher proportion of good solutions are searched more intensely. A GA starts with a group of randomly generated feasible solutions (referred to as populations) which evolve over multiple iterations (referred to as generations or trials) based on the principle of "survival of the fittest". The main procedures of a GA include encoding, selection, crossover, mutation and functional evaluation. This section will discuss each of the main procedures of the proposed GA, except the functional evaluation which will be discussed in detail in the next section. A flowchart describing the overall GA process is depicted in FIGURE 6.

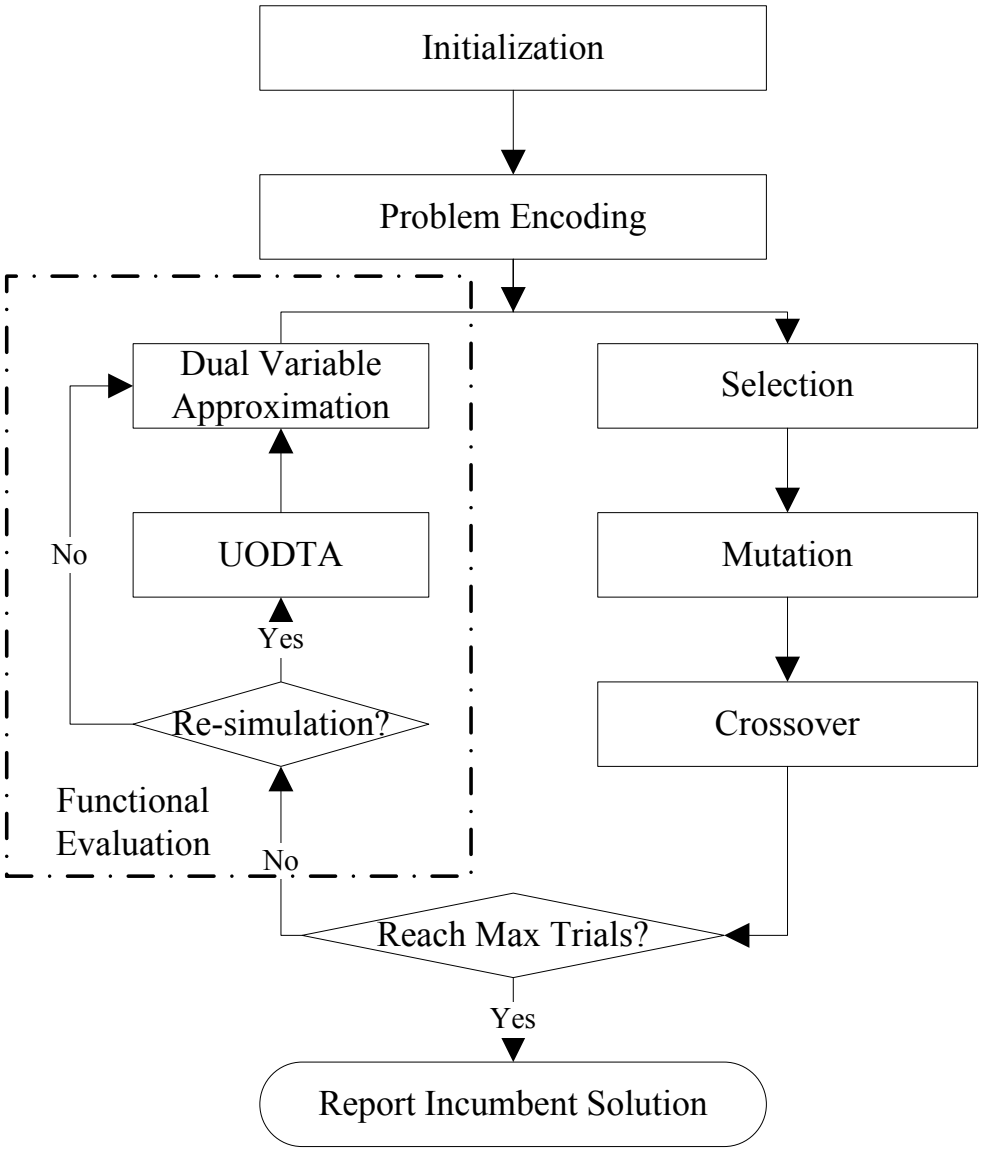

FIGURE 6: Proposed Genetic Algorithm

In the first step, the predefined parameters of GA, including the maximum number of functional evaluations, population size, crossover rate and mutation rate, are initialized. The next step is to decode the decision variables into binary form. In the BLPNDP studied in the dissertation, the decision variable b_i is continuous. However, GAs typically employ a binary string with a Lower Bound (LB) 0 and Upper Bound (UB) 1 to represent the variables. Thus, we address the floating point issue by applying

multiple bits for each b_i in the encoding step. The precision can be determined by the equation $2^{m-1} < \left((UB - LB) \times 10^{precision} + 1\right) \leq 2^m$. Using this equation, we choose 14 bits (m=14) to represent one b_i. The corresponding precision is 4, which is sufficient in this case.

In the selection stage, the Stochastic Universal Sampling (SUS) algorithm by Baker (1987) is employed to choose the new generation from the current generation based on the fitness measure. SUS selects the offspring with no bias and minimal spread while maintaining the same number of samples in a randomly shuffled manner. Mutation is applied after the selection step to explore a greater solution space. The mutation probability is a pre-defined parameter and is usually set to a low value to prevent the search from becoming a random search and to avoid genetic drift. A randomly chosen position of the structure is replaced with the random number of 0 or 1 when mutation occurs. Since Grefenstette (1990) indicates the theoretical advantage of two-point crossover when considering crossover operators, we adopt the two-point crossover and randomly choose two points in the structure and exchange the segments between the two crossover points.

A functional evaluation typically involves dynamic traffic simulation of time-dependent user optimal route flows for each feasible capacity expansion policy in the population in each generation. Because the traffic simulation based evaluation function is a computationally expensive, a dual variable approximation based evaluation function is developed to reduce the number of traffic simulations needed in this dissertation. The periodic re-simulation incorporated in the procedure is designed to periodically update the solution (x_i^t, y_{ij}^t) in the BLPNDP formulation so that the dual variable approximation can be more accurate. The following section discusses the details of estimating the evaluation function.

5.1 Evaluation Function

This section presents a novel evaluation function for the GA using dual variable approximation techniques to reduce the number of traffic simulations needed. In this dual approximation genetic algorithm, we employ the dual variable $\pi_i^{0,t} - \pi_i^{4,t}$ to design the evaluation function. The approximation techniques (backward connectivity algorithm and complimentary slackness conditions) of the dual variables are identical to those techniques presented in Chapter 4. Thus, we skip the details in this section and simply present the evaluation function design.

5.2 Design of the Evaluation Function

This section focuses on using the approximated dual variables to arrive at estimates of the evaluation function without re-simulation. As described earlier, $\pi_i^{0,t}$ is approximated by the TDSP travel time from cell *i* at time interval *t* to the destination. For the budget allocation to be effective, the value of $\pi_i^{0,t}$ should be as low as possible, which corresponds to a lower congestion level. Therefore, $\pi_i^{0,t}$ should be minimized and $\sum_i \sum_t \pi_i^{0,t}$ becomes the first part of the evaluation function.

$\pi_i^{2,t}$, $\pi_i^{3,t}$ and $\pi_i^{4,t}$ are the dual variables of the capacity-related constraints in BLPNDP. We can interpret $\pi_i^{2,t}$ as the change in total system travel time when the jam density of cell *i* at time *t* is changed by one unit (increased by one unit for consistency in this dissertation). For a budget allocation to be effective, the impact of the increase in jam density should not be significant. Let's consider an extreme case as an example. If a network is in free-flow condition, the increase in jam density will have

no impact on the total system travel time. Hence, the absolute value of $\pi_i^{2,t}$ should be minimized for budget allocation to be effective. The same reasoning can be applied to $\pi_i^{3,t}$ and $\pi_i^{4,t}$ as well. Therefore, the second part of the evaluation function is set to be equal to $\sum_i \sum_t \left(| \pi_i^{2,t} | + | \pi_i^{3,t} | + | \pi_i^{4,t} | \right)$. Note that the absolute operator is employed simply to construct a consistent objective function which minimizes the magnitude of dual variables.

Even though $\pi_i^{1,t}$ has no tangible meaning in this formulation, we choose to include it in the same manner as $\pi_i^{2,t}$, $\pi_i^{3,t}$ and $\pi_i^{4,t}$. Fortunately, from our preliminary results, $\pi_i^{1,t}$ is zero in most scenarios. To conclude, based on the approximated dual variables, the evaluation function for the proposed GA is $\sum_i \sum_t \left(\pi_i^{0,t} + | \pi_i^{1,t} | + | \pi_i^{2,t} | + | \pi_i^{3,t} | + | \pi_i^{4,t} | \right)$. The evaluation function completes the GA design for the stated problem. Next, we present the numerical experiments conducted with the proposed techniques.

5.3 Numerical Experiments

This section provides an overview of the various computational runs conducted along with the salient results. The objective of the computational runs is to demonstrate the computational savings obtained by using dual variable approximation techniques to estimate the evaluation function in GA versus the pure simulation based GA. We employ the GA designed by Grefenstette (1990) and make the necessary changes to serve the intended purpose. For this section, the GA based on the novel dual variable approximation techniques will be referred to as DAGA (Dual Approximation Genetic

Algorithm), while the conventional GA that employs DTA simulation as the evaluation function will be referred to as the PSGA (Pure Simulation Genetic Algorithm). Computational tests were conducted on three different networks of increasing size for various budget levels. Note that the DAGA also entails occasional traffic re-simulation to improve the accuracy of the dual variable approximations. For the computational experiments conducted, four different re-simulation frequencies are chosen to obtain UODTA flows: 1, 10, 25 and 50.

The following parameters are employed in both PSGA and DAGA: (i) number of functional evaluation 200,000, (ii) population size 500, (iii) crossover rate 0.6 and (iv) mutation rate 0.001. The crossover and mutation rate are the parameters calibrated in Karoonsoontawong and Waller (2006). The number of functional evaluation and population size are set to explore a broader solution space. The numerical experiments are conducted on a Linux machine with an Intel 1.86 GHz CPU and 2 GB memory. The DAGA, PSGA and dual variable approximation scheme are implemented in standard C programming language and compiled with a GNU Compiler Collection (GCC) compiler.

5.3.1. Experiment1 – 6-Cell CTM Network

In the first numerical experiment, we utilize the 6-cell CTM network depicted in FIGURE 2 to demonstrate the accuracy and efficiency of DAGA proposed in the work. In this experiment, the results are compared with the results from the modified K^{th}-Best algorithm and the PSGA. The modified K^{th}-best algorithm is implemented as in Karoonsoontawong (2006) in the C++ programming language with ILOG/CPLEX 9.0 Concert Technology Library. The algorithm systematically explores extreme points and ranks those vertices in the context of the Simplex method. The modified K^{th}-Best algorithm is an exact algorithm and can identify the global optimal solutions. However, the modified K^{th}-best algorithm is not suitable for network applications with realistic

sizes due to the bi-level nature of the BLPNDP and the exponential growth of the lower-level objective function (M_t to be precise).

The network characteristics and time-dependent OD demands are given below in TABLE 7 and TABLE 8, respectively.

TABLE 7: Characteristics of 6-cell CTM Network

Cell	N_i^t	Q_i^t
1	+inf	10
2	+inf	10
3	20	10
4	20	10
5	20	10
6	+inf	+inf

TABLE 8: Time-dependent OD Demands (d_i^t) for 6-cell CTM Network

	Origin	Destination	Number of Users
Time 1	1	6	30
	2	6	10
Time 2	1	6	20
	2	6	10

The demand level in this experiment is 70 vehicle trips, which are uniformly distributed over the planning horizon. The optimality gap was calculated with respect to the optimal solution obtained from the modified K^{th}-best algorithm. The results are summarized in TABLE 9. The numbers presented in the bracket for DAGA are the re-simulation frequency. For instance, DAGA (50) indicates that we re-simulate UODTA in every 50 trials.

TABLE 9: Numerical Results of 6-cell CTM Network with 70 vehicle trips

TAB	DAGA (50)			DAGA (25)			DAGA(10)			DAGA (1)			PSGA			Optimal Solution**
	TSTT	CPU Time*	Opt. Gap	TSTT	CPU Time*	Opt. Gap	TSTT	CPU Time*	Opt. Gap	TSTT	CPU Time*	Opt. Gap	TSTT	CPU Time*	Opt. Gap	TSTT
100	142	400.06	1.41%	140	1,021.06	0.00%	140	1,939.74	0.00%	140	16,163.18	0.00%	140	14,126.56	0.00%	140
90	140	708.07	0.00%	140	1,025.61	0.00%	140	1,874.13	0.00%	140	14,561.77	0.00%	140	14,057.87	0.00%	140
80	146	702.99	4.11%	141	1,020.40	0.71%	140	1,829.43	0.00%	140	14,137.16	0.00%	140	14,054.71	0.00%	140
70	157	378.25	10.83%	146	987.32	4.11%	144	1,814.63	2.86%	143	13,896.45	2.10%	140	14,506.94	0.00%	140
60	160	694.27	6.25%	158	988.67	5.06%	158	1,837.59	5.06%	152	13,963.71	1.32%	150	14,000.31	0.00%	150
50	163	711.37	1.84%	166	995.32	3.61%	164	1,803.89	2.50%	163	14,270.53	1.84%	160	14,117.52	0.00%	160
40	174	701.95	2.30%	174	1,008.47	2.30%	173	1,842.05	1.76%	172	14,129.41	1.16%	170	14,185.93	0.00%	170
30	196	617.86	8.16%	190	363.63	5.26%	184	1,831.43	2.17%	184	15,017.34	2.17%	180	14,054.55	0.00%	180
20	209	697.61	9.09%	208	519.90	8.65%	204	1,811.87	7.37%	202	14,133.14	5.94%	200	14,026.27	5.00%	190
10	233	166.16	9.87%	233	956.22	9.87%	233	1,796.32	9.87%	231	14,288.87	9.09%	230	14,157.77	8.70%	210
	Average Opt. Gap= 5.39%			Average Opt. Gap = 3.96%			Average Opt. Gap =3.16 %			Average Opt. Gap = 2.36%			Average Opt. Gap = 1.37%			

* in seconds

** from modified K^{th}-best algorithm

When re-simulation is conducted once in every 50 trials to update the UODTA solution (x_i^t, y_{ij}^t), the DAGA can obtain reasonable budget allocation with 1.17% (166.16 seconds versus 14,157.77 seconds when TAB = 10) to 5.04% (711.37 seconds versus 14,117.52 seconds when TAB = 50) of the CPU times required by PSGA. On average, DAGA (50) achieves an average optimality gap of 5.39% with an average computational savings of 96% across all budget levels. For DAGA (25), an average optimality gap of 3.96% was obtained at an average computational savings of 93.72 %. When the frequency of updating (x_i^t, y_{ij}^t) was increased to 1, the average optimality gap decreased to 2.36%. However, the CPU times increased significantly, since more computational effort is required. Though the PSGA performs well in terms of Total System Travel Time (TSTT), it is not possible to reduce the computational effort since every functional evaluation requires traffic simulation. On the other hand, DAGA provides an alternative evaluation function while maintaining solution quality and flexibility. The major advantage of the DAGA is that it eliminates the need for traffic simulation in each trial, especially when the number of time-dependent OD demands is large. In addition to the experiments conducted to show the accuracy of the DAGA, we further conduct numerical experiments on the same network with higher demand levels to demonstrate the computational efficiency of the proposed paradigm.

In the following experiment, we scale up the demand level to 700 vehicle trips and limit the allowed CPU times to approximately 600 CPU seconds for both DAGA and PSGA to simulate the situation when larger BLPNDP problems are to be solved when computational times available are limited. Note that it is difficult to construct the lower-level program for the modified K^{th}-best algorithm due to the exponential growth of the cost vector M_t. Furthermore, the NP-complete complexity makes solving the problem with the K^{th}-best algorithm computationally infeasible. Therefore, the

performance of DAGA and PSGA is compared at a similar number of functional evaluations by calculating the reduction in total system travel time with respect to PSGA. The vehicle trips are uniformly distributed over the planning horizon following the same demand pattern in TABLE 8 (i.e. at time 1, there are 30 and 10 vehicle trips departing from cell 1 and cell 2, respectively). We summarized the results in TABLE 10. On average, DAGA outperforms PSGA by 10.49% (DAGA 50), 10.60% (DAGA 25), 10.27% (DAGA 10) and 10.59% (DAGA 1) in terms of TSTT with higher demand level within the same CPU time. Marginally higher savings are obtained for lower budget levels when compared to the higher budget levels.

TABLE 10: Numerical Results of 6-cell CTM Network with 700 vehicle trips

TAB	DAGA (50) *		DAGA (25) *		DAGA (10) *		DAGA (1) *		PSGA *
	$TSTT_{50}$	$(TSTT_{50}-\tau)/\tau$	$TSTT_{25}$	$(TSTT_{25}-\tau)/\tau$	$TSTT_{10}$	$(TSTT_{10}-\tau)/\tau$	$TSTT_{1}$	$(TSTT_{1}-\tau)/\tau$	TSTT (τ)
100	2,320	-8.63%	2,322	-8.55%	2,320	-8.63%	2,316	-8.78%	2,539
90	2,319	-9.31%	2,360	-7.70%	2,330	-8.88%	2,325	-9.07%	2,557
80	2,334	-9.29%	2,338	-9.13%	2,374	-7.73%	2,364	-8.12%	2,573
70	2,405	-8.38%	2,399	-8.61%	2,405	-8.38%	2,390	-8.95%	2,625
60	2,426	-9.98%	2,422	-10.13%	2,408	-10.65%	2,426	-9.98%	2,695
50	2,465	-10.75%	2,442	-11.59%	2,487	-9.96%	2,441	-11.62%	2,762
40	2,522	-10.85%	2,470	-12.69%	2,515	-11.10%	2,493	-11.88%	2,829
30	2,540	-11.96%	2,536	-12.10%	2,538	-12.03%	2,535	-12.13%	2,885
20	2,549	-12.85%	2,555	-12.65%	2,559	-12.51%	2,560	-12.48%	2,925
10	2,582	-12.92%	2,584	-12.85%	2,584	-12.85%	2,584	-12.85%	2,965
	Average = -10.49%		Average = -10.60%		Average = -10.27%		Average = -10.59%		

* CPU Time allowed ≈ 600 seconds

5.3.2. Experiment 2 – 16-Cell CTM Network

To further test the DAGA, we employ the 16-cell CTM network (depicted in FIGURE 7) with 900 vehicle trips. The jam densities and saturation flow rates are shown in TABLE 11.

TABLE 11: Characteristics of 16-cell CTM Network

Cell	N_i^t	Q_i^t
1,2,3	+inf	8
16	+inf	+inf
others	10	8

Similarly, we limit the CPU times allowed to roughly 600 seconds. However, the vehicle trips are uniformly profiled into various time intervals to generate different congestion conditions.

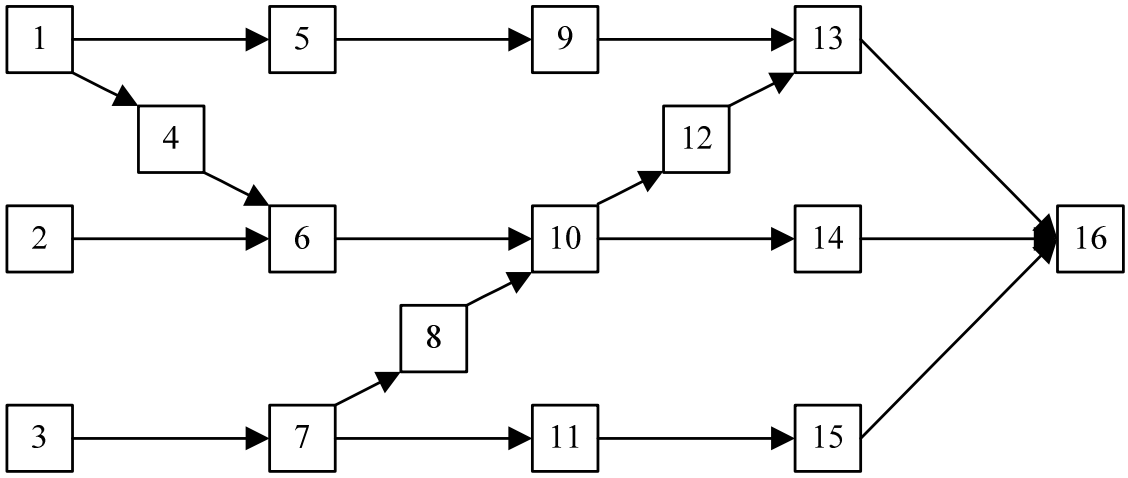

FIGURE 7: 16-cell CTM Network

When 900 vehicle trips are distributed uniformly over 45 and 30 time intervals, the departure rates are 20 and 30 vehicles per time interval, respectively. In this experiment, we limit the CPU time allowed to 600 seconds to obtain results in reasonable time. With the same jam densities and saturation flow rates, the congestion levels tend

to increase with the departure rates. From the summarized results in TABLE 12 and TABLE 13, we can observe that DAGA performs better when network is more congested. When compared to the total system travel time obtained under PSGA for the same number of functional evaluations, DAGA obtained an average reduction of up to 5.03% for DAGA (50) in the 45 time intervals experiment. For a total budget of 90 units, up to 16.16% reduction in total system travel time was observed. For most cases, DAGA still provided a reduction in TSTT for similar computational times. Note that DAGA (1) is presented simply to serve as a reference for comparison purposes. One should not employ the re-simulation frequency of one in DAGA as it requires extra work to approximate dual variables as compared to PSGA.

TABLE 12: Numerical Results of 16-cell CTM Network with 45 Time Intervals

TAB	DAGA (50)		DAGA (25)		DAGA (10)		DAGA (1)		PSGA
	$TSTT_{50}$	$(TSTT_{50}-\tau)/\tau$	$TSTT_{25}$	$(TSTT_{25}-\tau)/\tau$	$TSTT_{10}$	$(TSTT_{10}-\tau)/\tau$	$TSTT_{1}$	$(TSTT_{1}-\tau)/\tau$	TSTT (τ)
100	14,776	-10.29 %	16,828	2.17 %	16,090	-2.31 %	16,471	0.00 %	16,471
90	14,062	-16.16 %	17,721	5.65 %	15,382	-8.29 %	16,773	0.00 %	16,773
80	16,358	-5.20 %	18,288	5.99 %	17,381	0.73 %	17,255	0.00 %	17,255
70	15,650	-10.81 %	18,813	7.22 %	17,220	-1.86 %	17,546	0.00 %	17,546
60	16,492	-14.94 %	17,225	-11.16 %	18,604	-4.04 %	19,388	0.00 %	19,388
50	18,294	-6.28 %	19,498	-0.11 %	19,869	1.79 %	19,519	0.00 %	19,519
40	21,273	6.13 %	21,275	6.14 %	19,015	-5.13 %	20,928	4.41 %	20,044
30	22,746	3.33%	21,168	-3.83 %	21,275	-3.35 %	21,963	-0.22 %	22,012
20	22,785	3.87%	21,573	-1.65 %	21,990	0.25 %	21,963	0.12 %	21,936
10	23,580	0.00%	23,580	0.00 %	23,580	0.00 %	23,580	0.00 %	23,580
	-5.03%		1.04 %		-2.22 %		0.43 %		

TABLE 13: Numerical Results of 16-cell CTM Network with 30 Time Intervals

TAB	DAGA (50)		DAGA (25)		DAGA (10)		DAGA (1)		PSGA
	$TSTT_{50}$	$(TSTT_{50}-\tau)/\tau$	$TSTT_{25}$	$(TSTT_{25}-\tau)/\tau$	$TSTT_{10}$	$(TSTT_{10}-\tau)/\tau$	$TSTT_{1}$	$(TSTT_{1}-\tau)/\tau$	TSTT (τ)
100	17,682	-5.55 %	19,159	2.34 %	17,753	-5.17 %	18,721	0.00 %	18,721
90	18,984	-0.21 %	18,810	-1.12 %	17,632	-7.31 %	19,023	0.00 %	19,023
80	17,998	-7.73 %	18,276	-6.30 %	19,560	0.28 %	19,505	0.00 %	19,505
70	18,307	-7.52 %	21,227	7.23 %	19,470	-1.65 %	19,796	0.00 %	19,796
60	18,742	-13.38 %	20,336	-6.02 %	20,873	-3.54 %	21,638	0.00 %	21,638
50	22,253	2.22 %	22,115	1.59 %	22,119	1.61 %	21,769	0.00 %	21,769
40	20,843	-6.51 %	22,040	-1.14 %	21,742	-2.48 %	23,178	3.97 %	22,294
30	24,213	-0.20 %	22,909	-5.58 %	24,192	-0.29 %	24,213	-0.20 %	24,262
20	23,418	-3.18 %	24,213	0.11 %	24,240	0.22 %	24,213	0.11 %	24,186
10	25,830	0.00 %	25,830	0.00 %	25,830	0.00 %	25,830	0.00 %	25,830
	-4.20 %		-0.89 %		-1.83 %		0.39 %		

5.3.3. Experiment 3 – 68-Cell CTM Network

In the next numerical experiment, we employ a final network to show the potential impact of network size on the performance of DAGA and PSGA. The network contains 68 cells (depicted in FIGURE 3). With the data shown in

TABLE 4 and TABLE 5, we still allow 600 CPU seconds, $TAB = 50$ and 900 vehicle trips in this network. Because it takes significant computational time (135.28 hours to be exact) to obtain a reasonable solution with PSGA, we compare the solutions by setting limits on the CPU time. A re-simulation frequency of 10 was chosen for the DAGA because it performed reasonably in the previous experiments.

In this test, DAGA obtains the solution with the Total System Travel Time (TSTT) $\tau = 19{,}233$, while PSGA fails to complete the solution process within 600 CPU seconds. Hence, we reduce the number of vehicles in the network to 500 vehicle trips and increase the CPU time allowed to 6,000 seconds to obtain comparable solutions. With this setup, DAGA obtainss $\tau = 8{,}109$, while PSGA obtains $\tau = 8{,}724$, which shows an improvement of 7.05%. This percentage can be significant, especially when urban area network improvement is of interest. For instance, in our recent project funded by government agencies, the TSTT of the designated city is about 204,942 hours. The 7.05% improvement means 14,448 hours are saved with the proposed approach.

5.4 Summary

Genetic Algorithms, when applied to solve the BLPNDP, typically require dynamic traffic simulation as an evaluation function. The computational burden of traffic simulation prevents the usage of such an approach for networks of realistic sizes. This chapter examines the LP formulation of BLPNDP and devises an alternative solution of evaluating GA chromosomes by exploiting the specific CTM mathematical

structure. The developed evaluation function relies on the dual variable approximation techniques, including backward connectivity and complimentary slackness conditions. Based on the approximated dual variables, an estimate of the evaluation functions has been obtained which significantly reduces the number of traffic simulations needed. From the numerical results, it is evident that the GA with the proposed evaluation function yields stable solutions in a more computationally efficient manner. It should as well be noted that the proposed evaluation function is not restricted to GA. Many meta-heuristic approaches (i.e. simulated annealing, random search, tabu search) that require functional evaluation when solving the BLPNDP can utilize the evaluation function and gain considerable efficiency improvement.

Although the proposed GA solves the BLPNDP with great efficiency, it is not without its share of limitations. Most importantly, the dual variables employed in the evaluation function are approximated dual variables rather than exact ones. Further investigation of the dual variable approximation techniques is necessary to obtain the improved dual variable estimates.

Chapter 6. Single-destination Off-line Dynamic Traffic Assignment Capacity Calibration

Generally, transportation planning uses archived traffic data collected in the past; thus, it is considered an off-line application, whereas many management applications must be performed on-line using real-time data transmitted from an on-site surveillance system. In this dissertation, we focus on off-line DTA capacity calibration, in which the capacity employed in a DTA model is calibrated using counts data collected in the past. The off-line DTA capacity calibration problem requires adjusting the network capacities within a specified tolerance such that the difference between the DTA predictions and field observations can be minimized under user rationality assumptions. In other words, the capacity calibration formulation determines the capacity perturbation variables ($\bar{\chi}_i, \bar{\phi}_i$) to ensure that CTM model predictions match the actual field observations. The detailed formulation is presented below.

(I) Off-line DTA Capacity Calibration Formulation

$$\underset{x,y,\chi,\phi}{Min} \quad \sum_{i \in C \setminus C_S} \sum_{t \in T} | x_i^t - x_{i,a}^t | \tag{6.1}$$

subject to

$$\chi_{i,\min} \le \bar{\chi}_i \le \chi_{i,\max} \qquad \forall i \in C \setminus (C_R \cup C_s) \tag{6.2}$$

$$\phi_{i,\min} \le \bar{\phi}_i \le \phi_{i,\max} \qquad \forall i \in C \setminus C_s \tag{6.3}$$

$$\underset{x,y}{Min} \sum_{(i,j) \in E_S} \sum_{t \in T} \left(M_t \cdot y_{ij}^t \right) \tag{6.4}$$

$$x_i^t - x_i^{t-1} + \sum_{(i,j) \in FS(i)} y_{ij}^{t-1} - \sum_{(j,i) \in RS(i)} y_{ji}^{t-1} = d_i^t \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{0,t} \tag{6.5}$$

$$\sum_{(i,j)\in FS(i)} y_{ij}^t - x_i^t \le 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{1,t} \qquad (6.6)$$

$$\sum_{(j,i)\in RS(i)} y_{ji}^t \le \delta_i^t (N_i^t + \overline{\chi}_i - x_i^t) \qquad \forall i \in C \setminus C_s, t \in T \qquad : \pi_i^{2,t} \qquad (6.7)$$

$$\sum_{(j,i)\in RS(i)} y_{ji}^t \le Q_i^t + \overline{\phi}_i \qquad \forall i \in C \setminus C_s, t \in T \qquad : \pi_i^{3,t} \qquad (6.8)$$

$$\sum_{(i,j)\in FS(i)} y_{ij}^t \le Q_i^t + \overline{\phi}_i \qquad \forall i \in C \setminus C_s, t \in T \qquad : \pi_i^{4,t} \qquad (6.9)$$

$$x_i^0 = \zeta_i \qquad \forall i \in C \setminus C_S \qquad (6.10)$$

$$y_{ij}^0 = 0 \qquad \forall (i, j) \in E \qquad (6.11)$$

$$x_i^{|T|} = 0 \qquad \forall i \in C \setminus C_S \qquad (6.12)$$

$$x_i^t \ge 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad (6.13)$$

$$y_{ij}^t \ge 0 \qquad \forall (i, j) \in E, t \in T \qquad (6.14)$$

$$\overline{\chi}_i, \overline{\phi}_i \quad u.r.s. \qquad (6.15)$$

The upper-level objective function minimizes the sum of the absolute differences between cell occupancies predicted by the CTM model (x_i^t) and cell occupancies observed in the field ($x_{i,a}^t$). The upper-level program also specifies the bounds on the capacity decision variables ($\overline{\chi}_i$ and $\overline{\phi}_i$) to ensure that the perturbations are within an acceptable range (Eq. (6.2) and Eq. (6.3), respectively). Based on the relevant decision variables, cell jam density and saturation flow rate are calibrated according to Eqs. (6.7)-(6.9). Eq. (6.15) specifies that the decision variables are unrestricted-in-sign (u.r.s.), since the perturbation can potentially increase or decrease the capacity values on an as-needed basis. Lastly, similar to the network design formulation, CTM-based constraints (Eqs. (6.4), (6.5)-(6.6), (6.10)-(6.14)) are embedded to characterize both traffic dynamics and user behaviors. We note that, due to the different objective

function, the dual variables $\pi_i^{0,t}$ - $\pi_i^{4,t}$ have different implications in this formulation than those dual variables employed in the BLPNDP formulation even though the notations are identical.

By introducing a new variable, z_i^t, associated with each cell *i* at each time interval *t*, formulation *(I)* reduces to a bi-level linear program as follows:

(II) Reformulation of (I)

$$\underset{z,x,y,\chi,\phi}{Min} \quad \sum_{i\in C\setminus C_S}\sum_{t\in T} z_i^t \tag{6.16}$$

subject to

$$z_i^t \geq x_i^t - x_{i,a}^t \qquad \forall i \in C \setminus C_s, t \in T \tag{6.17}$$

$$z_i^t \geq x_{i,a}^t - x_i^t \qquad \forall i \in C \setminus C_s, t \in T \tag{6.18}$$

$$\chi_{i,\min} \leq \overline{\chi}_i \leq \chi_{i,\max} \qquad \forall i \in C \setminus (C_R \cup C_s) \tag{6.19}$$

$$\phi_{i,\min} \leq \overline{\phi}_i \leq \phi_{i,\max} \qquad \forall i \in C \setminus C_s \tag{6.20}$$

$$\underset{x,y}{Min} \sum_{(i,j)\in E_S}\sum_{t\in T}\left(M_t \cdot y_{ij}^t\right) \tag{6.21}$$

$$x_i^t - x_i^{t-1} + \sum_{(i,j)\in FS(i)} y_{ij}^{t-1} - \sum_{(j,i)\in RS(i)} y_{ji}^{t-1} = d_i^t \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{0,t} \tag{6.22}$$

$$\sum_{(i,j)\in FS(i)} y_{ij}^t - x_i^t \le 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{1,t} \qquad (6.23)$$

$$\sum_{(j,i)\in RS(i)} y_{ji}^t \le \delta_i^t (N_i^t + \overline{\chi}_i - x_i^t) \qquad \forall i \in C \setminus C_s, t \in T \qquad : \pi_i^{2,t} \qquad (6.24)$$

$$\sum_{(j,i)\in RS(i)} y_{ji}^t \le Q_i^t + \overline{\phi}_i \qquad \forall i \in C \setminus C_s, t \in T \qquad : \pi_i^{3,t} \qquad (6.25)$$

$$\sum_{(i,j)\in FS(i)} y_{ij}^t \le Q_i^t + \overline{\phi}_i \qquad \forall i \in C \setminus C_s, t \in T \qquad : \pi_i^{4,t} \qquad (6.26)$$

$$x_i^0 = \zeta_i \qquad \forall i \in C \setminus C_S \qquad (6.27)$$

$$y_{ij}^0 = 0 \qquad \forall (i,j) \in E \qquad (6.28)$$

$$x_i^{|T|} = 0 \qquad \forall i \in C \setminus C_S \qquad (6.29)$$

$$x_i^t \ge 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad (6.30)$$

$$y_{ij}^t \ge 0 \qquad \forall (i,j) \in E, t \in T \qquad (6.31)$$

$$z_i^t, \overline{\chi}_i, \overline{\phi}_i \quad u.r.s. \qquad (6.32)$$

The objective function of the new formulation minimizes the sum of the new variable z_i^t for all cells at all time intervals, other than sink cells. The variable z_i^t is set to be greater than the difference of the predicted occupancies x_i^t and the actual field occupancies $x_{i,a}^t$ of the cell (Eq. 6.17); in addition, z_i^t is greater than the difference

between the actual field occupancies $x_{i,a}^t$ and the predicted occupancy x_i^t of the cell (Eq. 6.18).

As mentioned before, the linear bi-level programming problem is known to be NP-complete. To address the computational complexity, we propose a Danzig-Wolfe decomposition-based heuristic to solve this problem efficiently. Though our method does not guarantee the optimality of the solution, a near-optimal solution can be found in reasonable computation time.

6.1 Danzig-Wolfe Based Decomposition Based Heuristic

In this section, we present the steps involved in developing the heuristic. Since there is not a known decomposition technique that can be applied directly to the specific bi-level program *(II)*, we adopt similar strategies described in Chapter 4 and first analyze the single-level linear program, which is essentially formulation *(II)* without the lower-level objective function (Eq. (6.21)). The user behaviors enforced by Eq. (6.21) will be incorporated in the later steps. We call this single-level program the *intermediate formulation*. For explanation purposes, some constraints are rearranged in the following formulation.

(III) Intermediate Single-level Formulation

$$\underset{x,y,\chi,\phi}{Min} \quad \sum_{i\in C\setminus C_S}\sum_{t\in T} z_i^t \tag{6.33}$$

subject to

$$z_i^t - x_i^t \geq -x_{i,a}^t \qquad \forall i \in C\setminus C_s, t\in T \qquad : \ \rho_i^{1,t} \tag{6.34}$$

$$z_i^t + x_i^t \geq x_{i,a}^t \qquad \forall i \in C\setminus C_s, t\in T \qquad : \rho_i^{2,t} \tag{6.35}$$

$$\bar{\phi}_i \geq \phi_{i,\min} \qquad \forall i \in C \setminus C_s \qquad : \rho_i^3 \qquad (6.36)$$

$$-\bar{\phi}_i \geq -\phi_{i,\max} \qquad \forall i \in C \setminus C_s \qquad : \rho_i^4 \qquad (6.37)$$

$$\bar{\chi}_i \geq \chi_{i,\min} \qquad \forall i \in C \setminus (C_R \cup C_s) \qquad : \rho_i^5 \qquad (6.38)$$

$$-\chi_i \geq -\chi_{i,\max} \qquad \forall i \in C \setminus (C_R \cup C_s) \qquad : \rho_i^6 \qquad (6.39)$$

$$x_i^t - x_i^{t-1} + \sum_{(i,j)\in FS(i)} y_{ij}^{t-1} - \sum_{(j,i)\in RS(i)} y_{ij}^{t-1} = d_i^t \qquad \forall i \in C \setminus C_s, t \in T \qquad : \pi_i^{0,t} \qquad (6.40)$$

$$-\sum_{(i,j)\in FS(i)} y_{ij}^t + x_i^t \geq 0 \qquad \forall i \in C \setminus C_s, t \in T \qquad : \pi_i^{1,t} \qquad (6.41)$$

$$-\sum_{(j,i)\in RS(i)} y_{ji}^t + \delta_i^t \chi_i - \delta_i^t x_i^t \geq -\delta_i^t N_i^t \qquad \forall i \in C \setminus C_s, t \in T \qquad : \pi_i^{2,t} \qquad (6.42)$$

$$-\sum_{(j,i)\in RS(i)} y_{ji}^t + \phi_i \geq -Q_i^t \qquad \forall i \in C \setminus C_s, t \in T \qquad : \pi_i^{3,t} \qquad (6.43)$$

$$-\sum_{(i,j)\in FS(i)} y_{ij}^t + \phi_i \geq -Q_i^t \qquad \forall i \in C \setminus C_s, t \in T \qquad : \pi_i^{4,t} \qquad (6.44)$$

$$x_i^0 = \zeta_i \qquad \forall i \in C \setminus C_S \qquad (6.45)$$

$$y_{ij}^0 = 0 \qquad \forall (i,j) \in E \qquad (6.46)$$

$$x_i^{|T|} = 0 \qquad \forall i \in C \setminus C_S \qquad (6.47)$$

$$x_i^t \geq 0 \qquad \forall i \in C \setminus C_s, t \in T \qquad (6.48)$$

$$y_{ij}^t > 0 \qquad \forall (i,j) \in E, t \in T \qquad (6.49)$$

$$z_i^t, \chi_i, \phi_i \quad u.r.s. \qquad (6.50)$$

The single-level formulation attempts to find the user flows and network capacities that minimize the absolute difference between the cell occupancies predicted by the CTM model and the actual field occupancies when the users are not constrained to reach the destination at the earliest possible time. In other words, the assumption that users are selfish is temporarily relaxed in this formulation. The dual variable corresponding to each constraint is indicated on the right side of the constraint (ρ's and π's). We then choose to apply the Dantzig-Wolfe decomposition principle to the dual formulation of *(III)*. The dual formulation of *(III)* is given below:

(IV) Dual Formulation of (III)

$$\underset{\pi,\rho}{Max} \quad \sum_{t\in T}\sum_{i\in C_R} d_i^t \pi_i^{0,t} - \sum_{t\in T}\sum_{i\in C\setminus(C_R\cup C_S)} \delta_i^t N_i^t \pi_i^{2,t} - \sum_{t\in T}\sum_{i\in C\setminus(C_R\cup C_S)} Q_i^t \pi_i^{3,t} - \sum_{t\in T}\sum_{i\in C\setminus C_S} Q_i^t \pi_i^{4,t} +$$

$$\sum_{i\in C\setminus C_S}\sum_{t\in T}(\rho_i^{2,t}-\rho_i^{1,t})x_{i,a}^t + \sum_{i\in C\setminus C_S}\rho_i^3\phi_{i,\min} - \sum_{i\in C\setminus C_S}\rho_i^4\phi_{i,\max} + \sum_{i\in C\setminus(C_R\cup C_S)}\rho_i^5\chi_{\min} - \sum_{i\in(C_R\cup C_S)}\rho_i^6\chi_{\max} \tag{6.51}$$

subject to

$$\rho_i^3 - \rho_i^4 + \sum_{t\in T}\left(\pi_i^{3,t}+\pi_i^{4,t}\right) = 0 \qquad \forall i\in C\setminus C_S \qquad : \bar{\phi}_i \tag{6.52}$$

$$\rho_i^5 - \rho_i^6 + \sum_{t\in T}\delta_i^t\pi_i^{2,t} = 0 \qquad \forall i\in C\setminus(C_R\cup C_S) \qquad : \bar{\chi}_i \tag{6.53}$$

$$\rho_i^{2,t} + \rho_i^{1,t} = 1 \qquad \forall i\in C\setminus C_S, t\in T \qquad : z_i^t \tag{6.54}$$

$$-\rho_i^{1,t} + \rho_i^{2,t} + \pi_i^{0,t} - \pi_i^{0,t+1} + \pi_i^{1,t} \le 0 \qquad \forall i\in C_R, t\in T\setminus\{|T|\} \qquad : x_i^t \tag{6.55}$$

$$-\rho_i^{1,t} + \rho_i^{2,t} + \pi_i^{0,t} - \pi_i^{0,t+1} + \pi_i^{1,t} - \delta_i^t\pi_i^{2,t} \le 0 \qquad \forall i\in C\setminus(C_R\cup C_S), t\in T\setminus\{|T|\}$$

$$: x_i^t \tag{6.56}$$

$$\pi_i^{0,t+1} - \pi_j^{0,t+1} - \pi_i^{1,t} - \pi_j^{2,t} - \pi_j^{3,t} - \pi_i^{4,t} \le 0 \quad \forall (i,j) \in E \setminus E_S, t \in T \setminus \{|T|\} \quad : y_{ij}^t \quad (6.57)$$

$$\pi_i^{0,t+1} - \pi_i^{1,t} - \pi_i^{4,t} \le 0 \quad \forall (i,j) \in E_S, t \in T \setminus \{|T|\} \quad : y_{ij}^t \quad (6.58)$$

$$\pi_i^{1,t}, \pi_i^{2,t}, \pi_i^{3,t}, \pi_i^{4,t} \ge 0, \quad \pi_i^{0,t} \quad u.r.s \quad (6.59)$$

$$\rho_i^{1,t}, \rho_i^{2,t}, \rho_i^3, \rho_i^4, \rho_i^5, \rho_i^6 \ge 0 \quad (6.60)$$

We apply the Dantzig-Wolfe decomposition principle to *(IV)* such that the resulting restricted master problem is a Linear Program (LP) and the resulting pricing problem is the dual formulation of the DTA with the objective function ($\underset{x,y}{Min} \sum_{t \in T} \sum_{(i,j) \in E_S} 0 \times y_{ij}^t$), which is potentially impractical. Since the decomposed pricing problem has an unrealistic objective function and the lower-level program is in fact UODTA, we heuristically replace the pricing problem by UODTA to account for the user behaviors. UODTA then can be solved to optimality by the existing combinatorial algorithm (Waller and Ziliaskopoulos, 2006). The decomposed restricted master and pricing problem are presented below:

(V) Dantzig-Wolfe Restricted Master Program:

$$\underset{w_v, \rho}{Max}$$

$$\sum_{v=1}^{V} \left\{ w_v \sum_{t \in T} \left[\sum_{i \in C_R} d_i^t \pi_i^{0,t,v} - \sum_{i \in C \setminus (C_R \cup C_S)} \left(\delta_i^t N_i^t \pi_i^{2,t,v} + Q_i^t \pi_i^{3,t,v} \right) - \sum_{i \in C \setminus C_S} Q_i^t \pi_i^{4,t,v} \right] \right\}$$
$$\sum_{i \in C \setminus C_S} \sum_{t \in T} \left(\rho_i^{2,t} - \rho_i^{1,t} \right) x_{i,a}^t + \sum_{i \in C \setminus C_S} \rho_i^3 \phi_{i,\min} - \sum_{i \in C \setminus C_S} \rho_i^4 \phi_{i,\max} + \sum_{i \in C \setminus (C_R \cup C_S)} \rho_i^5 \chi_{\min} - \sum_{i \in C \setminus (C_R \cup C_S)} \rho_i^6 \chi_{\max} \quad (6.61)$$

subject to

$$\rho_i^3 - \rho_i^4 + \sum_{v=1}^{V} w_v \left[\sum_{t \in T} \left(\pi_i^{3,t,v} + \pi_i^{4,t,v} \right) \right] = 0 \qquad \forall i \in C \setminus C_S \qquad : \bar{\phi}_i \qquad (6.62)$$

$$\rho_i^5 - \rho_i^6 + \sum_{v=1}^{V} w_v \left(\sum_{t \in T} \delta_i^t \pi_i^{2,t,v} \right) = 0 \qquad \forall i \in C \setminus (C_R \cup C_S) \qquad : \bar{\chi}_i \qquad (6.63)$$

$$\rho_i^{2,t} + \rho_i^{1,t} = 1 \qquad \forall i \in C \setminus C_S, t \in T \qquad : z_i^t \qquad (6.64)$$

$$\sum_{v=1}^{V} w_v = 1 \qquad : g \qquad (6.65)$$

$$\rho_i^{1,t}, \rho_i^{2,t}, \rho_i^3, \rho_i^4, \rho_i^5, \rho_i^6 \geq 0 \qquad (6.66)$$

$$w_v \geq 0 \quad \forall v = 1,2,...,V \qquad (6.67)$$

(VI) Dantzig-Wolfe Pricing Program:

$$\underset{\pi_i^{0,t}, \pi_i^{1,t}, \pi_i^{2,t}, \pi_i^{3,t}, \pi_i^{4,t}}{Max} \sum_{t \in T} \left(\sum_{i \in C_R} d_i^t \pi_i^{0,t} - \sum_{i \in C \setminus (C_R \cup C_S)} \delta_i^t N_i^t \pi_i^{2,t} - \sum_{i \in C \setminus (C_R \cup C_S)} Q_i^t \pi_i^{3,t} - \sum_{i \in C \setminus C_S} Q_i^t \pi_i^{4,t} \right)$$

$$- \sum_{i \in C \setminus C_S} \bar{\phi}_i \left(\sum_{t \in T} \pi_i^{3,t} + \pi_i^{4,t} \right) - \sum_{i \in C \setminus (C_R \cup C_S)} \bar{\chi}_i \left(\sum_{t \in T} \delta_i^t \pi_i^{2,t} \right) - \sum_{i \in C \setminus C_S} \sum_{t \in T} z_i^t - g \qquad (6.68)$$

subject to

$$-\rho_i^{1,t} + \rho_i^{2,t} + \pi_i^{0,t} - \pi_i^{0,t+1} + \pi_i^{1,t} \leq 0 \qquad \forall i \in C_R, t \in T \qquad (6.69)$$

$$-\rho_i^{1,t} + \rho_i^{2,t} + \pi_i^{0,t} - \pi_i^{0,t+1} + \pi_i^{1,t} - \delta_i^t \pi_i^{2,t} \leq 0 \qquad \forall i \in C \setminus (C_R \cup C_S), t \in T \setminus \{|T|\}$$

$$(6.70)$$

$$\pi_i^{0,t+1} - \pi_j^{0,t+1} - \pi_i^{1,t} - \pi_j^{2,t} - \pi_j^{3,t} - \pi_i^{4,t} \le 0 \qquad \forall (i,j) \in E \setminus E_S, t \in T \setminus \{|T|\} \qquad (6.71)$$

$$\pi_i^{0,t+1} - \pi_i^{1,t} - \pi_i^{4,t} \le 0 \qquad \forall (i,j) \in E_S, t \in T \setminus \{|T|\} \qquad (6.72)$$

$$\pi_i^{1,t}, \pi_i^{2,t}, \pi_i^{3,t}, \pi_i^{4,t} \ge 0 \qquad \pi_i^{0,t} \quad u.r.s. \qquad (6.73)$$

The master problem consists of constraints whose dual variables are the upper-level decision vectors ($\overline{\chi}, \overline{\phi}, z$), while the pricing problem contains the constraints whose dual variables are the lower-level decision vectors. The restricted master problem has a fixed number of constraints. However, a new variable w_v is augmented to the program that corresponds to the new extreme point generated from the pricing problem in each iteration. Thus, the decision vector of the master problem increases with the number of iterations. In other words, the pricing problem generates a new extreme point in each iteration, while the restricted master problem determines the optimal convex combination of the generated extreme points. The restricted master problem is a LP and can be solved by virtually any LP solver. The pricing problem is the UODTA with modified capacities and can be solved by an existing combinatorial algorithm. The Dantzig-Wolfe decomposition-based heuristic scheme is summarized in FIGURE 8.

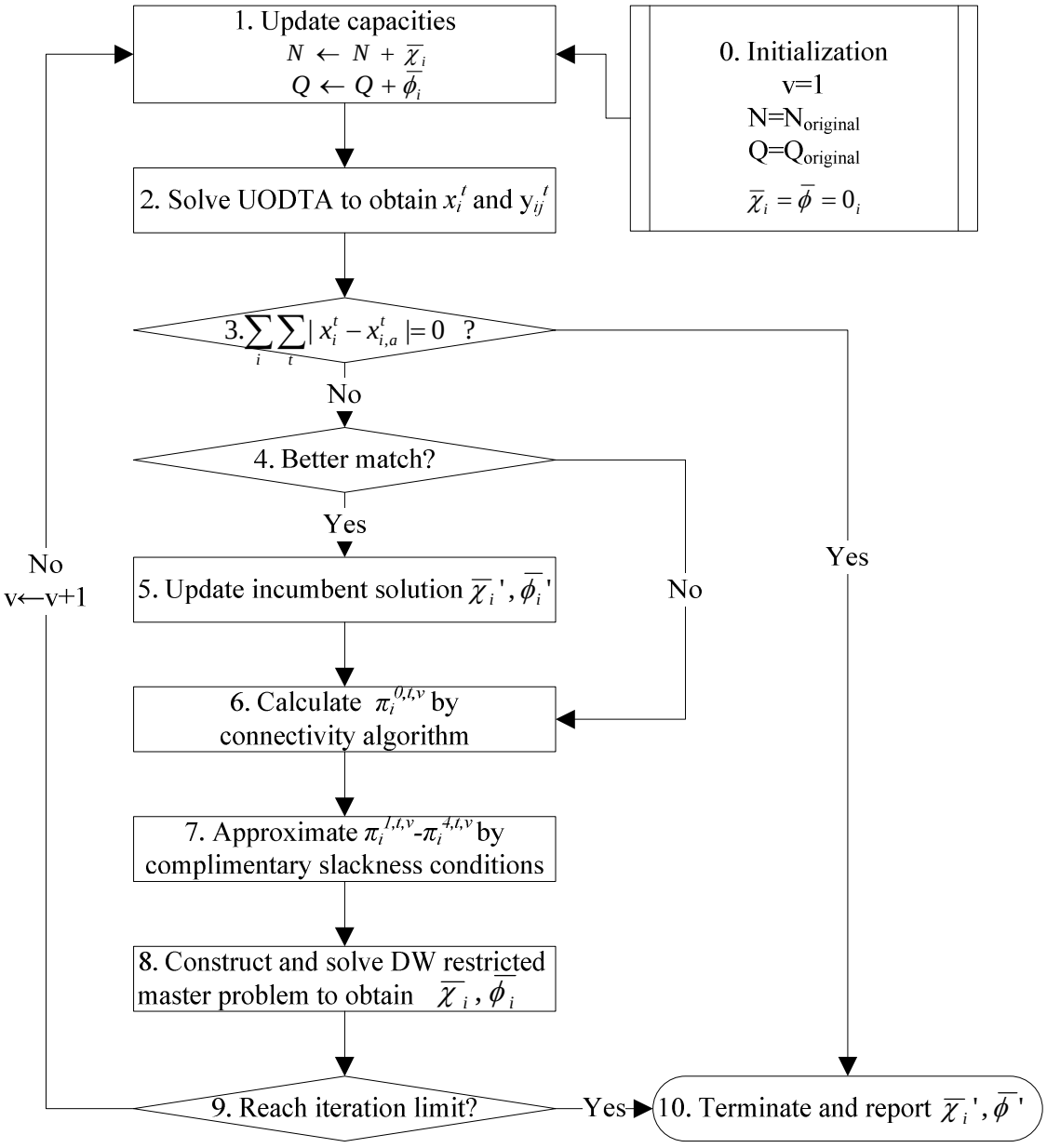

FIGURE 8: Dantzig-Wolfe Decomposition based Heuristic

Initially, the number vertex v=1, jam density $N = N_{original}$, saturation flow rate $Q = Q_{original}$ and flow perturbations $\bar{\chi}_i = \bar{\phi}_i = 0$. The network capacities are modified according to ($\bar{\chi}_i$,$\bar{\phi}_i$) in Step 1. The heuristic scheme then applies the combinatorial

algorithm to obtain UODTA (x_i^t, y_{ij}^t) in Step 2. If the predicted occupancies and actual field occupancies match within a reasonable gap ($\left\{ \sum_{i \in C \setminus C_S} \sum_{t \in T} | x_i^t - x_{i,a}^t | \right\} / \sum_{i \in C \setminus C_S} \sum_{t \in T} x_{i,a}^t \leq 1\%$) in Step 3, the procedure stops and the incumbent solutions ($\bar{\chi}_i^{'}, \bar{\phi}_i^{'}$) are reported. Otherwise, the process continues. In Step 4, the heuristic checks the count match. If the current ($\bar{\chi}_i, \bar{\phi}_i$) yields a better count match, the procedure updates the incumbent solutions in Step 5; otherwise, the process goes to Step 6. Dual variables $\pi_i^{k,t,v} \ \forall k \in \{0,1,2,3,4\}$ can be obtained by the connectivity algorithm and complimentary slackness conditions based on the UODTA (x_i^t, y_{ij}^t) in Step 7. The details of the connectivity algorithm and complimentary slackness conditions will be discussed in the next section. The Dantzig-Wolfe restricted master problem is then constructed using the approximated dual variables and solved by an LP solver in Step 8. Consequently, capacity perturbations ($\bar{\chi}_i, \bar{\phi}_i$) can be acquired by reading the dual variables associated with the constraints in the restricted master problem. The process repeats until it reaches the iteration limit or the counts match.

6.2 Dual Variable Approximation

Dual variables play a crucial role in the decomposition scheme. Therefore, we devise a combinatorial algorithm and utilize the complimentary slackness conditions to efficiently approximate the dual variables needed. We note that an existing combinatorial algorithm is applied to find the UODTA solutions (Step 2 in FIGURE 8) before dual variable approximation. Hence, x_i^t and y_{ij}^t are exogenous for the following

dual variable approximation procedure. Next, we discuss the details of the connectivity algorithm for dual variable π_i^0.

6.2.1. Connectivity Algorithm for Dual Variable π_i^0 Approximation

The algorithm proposed in this section works on a time-expanded CTM network, which is the expansion of a CTM network into planning horizon (*T*) while maintaining connectivity. Examples of the original and time-expanded network are given in FIGURE 2 and FIGURE 9, respectively.

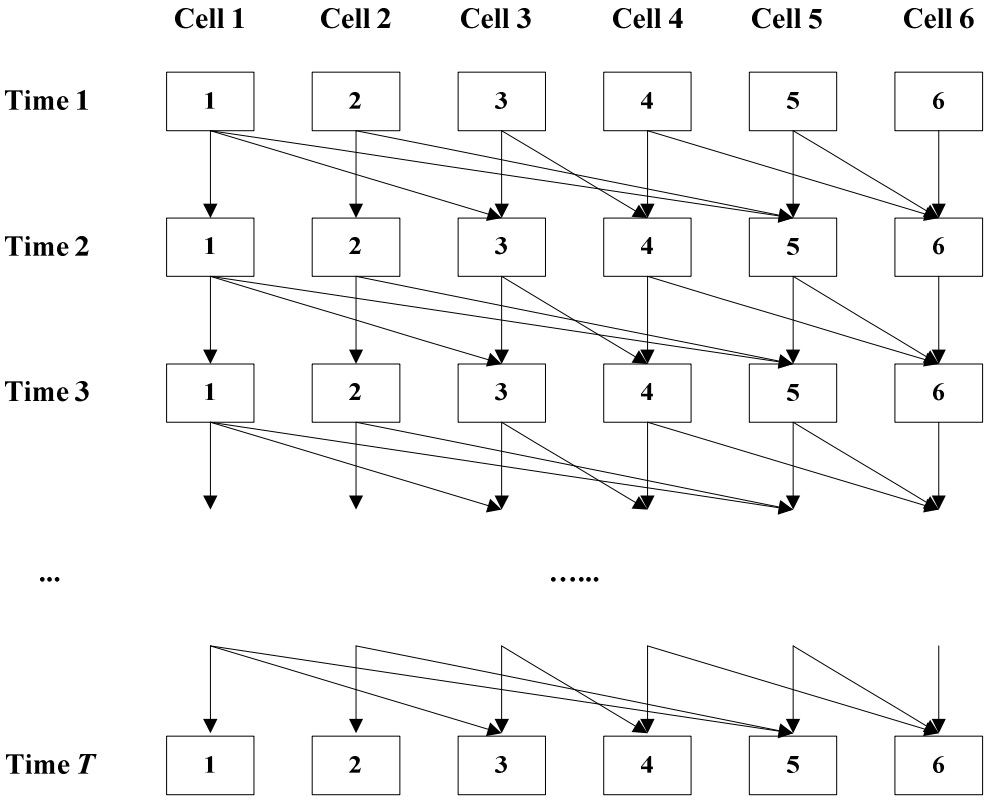

FIGURE 9: Example of time-expanded network

In this connectivity algorithm, three labels are kept for each cell. The first label *diff(i,t)* indicates the associations between the predicted occupancies x_i^t and actual field

occupancies $x_{i,a}^t$ of cell *i* at time interval *t*. If x_i^t of a cell exceeds $x_{i,a}^t$, *diff(i,t)* of that cell is initialized to one. Otherwise, *diff(i,t)* is initialized to negative one. This initialization is used because, as an additional vehicle passes through a cell, $(x_i^t - x_{i,a}^t)$ is increased by one unit if $x_i^t \geq x_{i,a}^t$. Otherwise, the addition of the vehicle reduces $(x_i^t - x_{i,a}^t)$ by one unit. The second label, *dist(i,t)*, gives the length of the time-dependent shortest path from cell *i* at time interval *t* to the destination cell. The distance label, *dist(i,t)*, is initialized to infinity. The third label, *downstream(i,t)*, stores the downstream cell of cell *i* at time interval *t* in the Time-Dependent Shortest Path (TDSP). The label is initialized with *(INF,INF)*.

A Scan Eligible (SE) list is employed to update the labels. The initial SE list consists of the sink cells at different time intervals. The insertion of the sink cells into the SE list is completed in decreasing order of time. For instance, the sink cell at time interval *T* is inserted before the sink cell at time interval *T – 1* is added. The first cell *k* in the SE list is selected and all the connectors to the selected cell are analyzed. Let the current cell be *(j,t)* and upstream cell be *(i,t-1)*. If the connector is not capacitated, the upstream cell is not capacitated and *dist(i,t-1)>dist(j,t)*+1, then *dist(i,t-1)=dist(j,t)*+1 and *downstream(i,t-1)=(j,t)*. The updated cell is added to SE and the selected cell *k* is then deleted from the SE. The process repeats until the SE list is empty. Using the finalized downstream label, the process traces the TDSP to its destination and calculates the dual variable π_i^0 by summing the *diff(i,t)* labels along that path. To be precise, we present the pseudo-code:

Step 0: Initialization

If $x_i^t \geq x_{i,a}^t$, then *diff(i,t)* = 1 $\quad \forall i \in C \setminus C_S, t \in T$

Otherwise *diff(i,t)* = -1 $\quad \forall i \in C \setminus C_S, t \in T$

$downstream(i,t) = (INF, INF)$ $\quad \forall i \in C \setminus C_S, t \in T$

$dist(i,t) = \infty$ $\quad \forall i \in C \setminus C_S, t \in T$

$\pi_i^{0,t} = 0$ $\quad \forall i \in C \setminus C_S, t \in T$

Step 1: Label updating

Insert $\{(j,t), \forall j \in C_s, t \in T\}$ into SE in a decreasing order of time

While SE is not empty

Remove *(j,t)* from the front of SE

$\forall i \in \Gamma^{-1}(j), i \in C \setminus C_S$

If $N_i^{t-1} - x_i^{t-1} > 0$, $Q_{ij}^{t-1} - \sum_{(i,j) \in FS(i)} y_{ij}^{t-1} > 0$ and *dist(i,t-1) > dist(j,t)*+1

dist(i,t-1) = dist(j,t)+1

downstream(i,t-1) = (j,t);

Insert *(i,t-1)* into the SE

End if

End while

Step2: Dual variable calculation

For $i \in C \setminus C_S, t \in T$

$(i', t') = downstream(i, t)$

While (i', t') is not a sink cell

$\pi_i^{0,t} = \pi_i^{0,t} + diff\,(i', t')$

$(i', t') = downstream(i', t')$

End while

End for

The proposed connectivity algorithm is able to find the dual variable π_i^0 with simple label updating. From the preliminary experiments, the algorithm finds π_i^0 with great efficiency and facilitates the usage of π_i^0 in approximating the rest of the required dual variables.

6.2.2. Complimentary Slackness Conditions for $\pi_i^{1,t}, \pi_i^{2,t}, \pi_i^{3,t}, \pi_i^{4,t}$ Approximation

After obtaining UODTA solutions (x_i^t, y_{ij}^t) and $\pi_i^{0,t}$, we develop dual variable approximation techniques to efficiently calculate the rest of the dual variables to construct the Dantzig-Wolfe decomposition restricted master problem. It should be noted that the values of the *x's, y's, π's* and *ρ's* are known before the process begins in this section. The dual approximation techniques utilize the complementary slackness conditions from both the primal and dual formulation as follows.

Non-sink cells

Complimentary Slackness Condition

If $\sum_{(i,j)\in FS(i)} y_{ij}^t - x_i^t < 0$ then set $\pi_i^{1,t} = 0,\ \forall i \in C \setminus C_S, t \in T$ (from Eq. (6.41))

If $\sum_{(i,j)\in RS(i)} y_{ij}^t < \delta_i^t (N_i^t + \chi_i - x_i^t)$ then set $\pi_i^{2,t} = 0,\ \forall i \in C \setminus C_S, t \in T$

(from Eq. (6.42))

If $\sum_{(i,j)\in RS(i)} y_{ij}^t < Q_i^t + \phi_i$ then set $\pi_i^{3,t} = 0,\ \forall i \in C \setminus C_S, t \in T$ (from Eq. (6.43))

If $\sum_{(i,j)\in FS(i)} y_{ij}^t < Q_i^t + \phi_i$ then set $\pi_i^{4,t} = 0,\ \forall i \in C \setminus C_S, t \in T$ (from Eq. (6.44))

Source Cells

Complimentary Slackness Condition (from Eq. (6.55))

If $x_i^t > 0$ then $\pi_i^{1,t} = \pi_i^{0,t+1} - \pi_i^{0,t} + \rho_i^{1,t} - \rho_i^{2,t}$

else approximate $\pi_i^{1,t} = \pi_i^{0,t+1} - \pi_i^{0,t} + \rho_i^{1,t} - \rho_i^{2,t}$

Non-source & non-sink cells

Complimentary Slackness Condition (from Eq. (6.56))

If $x_i^t > 0$, set $-\rho_i^{1,t} + \rho_i^{2,t} + \pi_i^{0,t} - \pi_i^{0,t+1} + \pi_i^{1,t} - \delta_i^t \pi_i^{2,t} = 0$

Else approximate $-\rho_i^{1,t} + \rho_i^{2,t} + \pi_i^{0,t} - \pi_i^{0,t+1} + \pi_i^{1,t} - \delta_i^t \pi_i^{2,t} = 0$

Detailed Calculation

If $\pi_i^{1,t}$ is determined and $\pi_i^{2,t}$ is undetermined

Then $\pi_i^{2,t} = \left(\pi_i^{1,t} - \pi_i^{0,t+1} + \pi_i^{0,t} - \rho_i^{1,t} + \rho_i^{2,t}\right) \Big/ \delta_i^t$

Else if $\pi_i^{2,t}$ is determined and $\pi_i^{1,t}$ is undetermined

Then $\pi_i^{1,t} = \delta_i^t \pi_i^{2,t} + \rho_i^{1,t} - \rho_i^{2,t} - \pi_i^{0,t} + \pi_i^{0,t+1}$

Else

Assume $\pi_i^{1,t} = \pi_i^{2,t} = \left(-\pi_i^{0,t+1} + \pi_i^{0,t} - \rho_i^{1,t} + \rho_i^{2,t}\right) \Big/ 2 \times \delta_i^t$

Non-sink cell connector

Complimentary Slackness Condition (from Eq. (6.57))

If $y_{ij}^t > 0$, set $\pi_i^{0,t+1} - \pi_j^{0,t+1} - \pi_i^{1,t} - \pi_j^{2,t} - \pi_j^{3,t} - \pi_i^{4,t} = 0$

Else approximate $\pi_i^{0,t+1} - \pi_j^{0,t+1} - \pi_i^{1,t} - \pi_j^{2,t} - \pi_j^{3,t} - \pi_i^{4,t} = 0$

Detailed Calculation

If $\pi_j^{3,t}$ is determined and $\pi_i^{4,t}$ is undetermined

Then $\pi_i^{4,t} = \pi_i^{0,t+1} - \pi_j^{0,t+1} - \pi_i^{1,t} - \pi_j^{2,t} - \pi_j^{3,t}$

Else if $\pi_i^{4,t}$ is determined and $\pi_j^{3,t}$ is undetermined

Then $\pi_j^{3,t} = \pi_i^{0,t+1} - \pi_j^{0,t+1} - \pi_i^{1,t} - \pi_j^{2,t} - \pi_i^{4,t}$

Else

Assume $\pi_j^{3,t} = \pi_i^{4,t} = (\pi_i^{0,t+1} - \pi_j^{0,t+1} - \pi_i^{1,t} - \pi_j^{2,t}) / 2$

Sink cell connector

Complimentary Slackness Condition (from Eq. (6.58))

If $y_{ij}^t > 0$, set $\pi_i^{0,t+1} - \pi_i^{1,t} - \pi_i^{4,t} = 0$

Else approximate $\pi_i^{0,t+1} - \pi_i^{1,t} - \pi_i^{4,t} = 0$

Detailed Calculation

If $\pi_i^{1,t}$ is determined and $\pi_i^{4,t}$ is undetermined

Then $\pi_i^{4,t} = \pi_i^{0,t+1} - \pi_i^{1,t}$

Else if $\pi_i^{4,t}$ is determined and $\pi_i^{1,t}$ is undetermined

Then $\pi_i^{1,t} = \pi_i^{0,t+1} - \pi_i^{4,t}$

Else

Assume $\pi_i^{1,t} = 0$ and $\pi_i^{4,t} = \pi_i^{0,t+1} - \pi_i^{1,t}$

To summarize, the proposed heuristic contains three primary components: the restricted master problem, the pricing problem and the dual variable approximation. The restricted master problem finds the values of ($\bar{\chi}, \bar{\phi}, \rho$) and passes them to the pricing problem, which is heuristically replaced by UODTA. The pricing problem is solved to obtain the occupancies and flows (*x*,*y*) and passes the values to the dual approximation procedures. The dual approximation procedures approximate the dual variables (π) and pass them back to the restricted master problem to augment its column. The process repeats until a stopping criterion is met.

6.3 Numerical Experiments

In the numerical experiments, we first employ the 6-cell CTM network depicted in FIGURE 2 to demonstrate the accuracy and efficiency of the proposed heuristic scheme. The running environment is Windows XP with an Intel 2.80 GHz CPU and 2 GB memory. LP Solver MINOS (Murtagh and Saunders, 1998) is used to solve the Dantzig-Wolfe restricted master problem. The time-dependent OD demands and network characteristics are given in TABLE 14 and TABLE 15, respectively.

TABLE 14: Time-dependent OD Demands for 6-cell CTM Network

	Origin	Destination	Demand
Time 1	1	6	1
	2	6	1
Time 2	1	6	2
	2	6	2
Time 3	2	6	1

TABLE 15: Characteristics of 6-cell CTM Network

Cell	Actual N_i^t	Actual Q_i^t
1	+inf	3
2	+inf	3
3	1	3
4	1	3
5	3	3
6	+inf	+inf

To obtain the target counts, we first run the user optimal dynamic traffic assignment (UODTA) using the network data in TABLE 15. The cell occupancies $x_{i,a}^t$ are calculated based on the UODTA to simulate the "real-world" occupancies/counts that the heuristic will match. Next, we assume that we are given the randomly generated non-calibrated N_i^t and Q_i^t in TABLE 16 and $x_{i,a}^t$ from the procedure mentioned above, and then we start the capacity calibration.

TABLE 16: Calibrated Results

Cell	Non-calibrated N_i^t	Non-calibrated Q_i^t	Calibrated N_i^t	Calibrated Q_i^t
1	+inf	1	+inf	3
2	+inf	5	+inf	3
3	3	1	1	3
4	3	1	1	3
5	5	1	3	3
6	+inf	+inf	+inf	+inf

In this experiment, the proposed Dantzig-Wolfe decomposition-based heuristic scheme predicts flows that perfectly match the count within one minute of computation time. Through this experiment, it is observed that a non-unique solution with the same objective value could potentially exist. For instance, if the calibrated Q_i^t in cell 2 is 4 instead of 3, it is possible that the flows and counts still match due to the redundant capacity in that cell. However, as stated in the mathematical formulation, it is the counts that the proposed heuristic is trying to match. In addition, the non-uniqueness is bounded by the perturbation constraints in Eq. (6.2) and Eq. (6.3).

In the next experiment, we generate the target counts using the same procedure applied in the previous experiment and randomly generate the capacities for a 68-cell CTM network (FIGURE 3). The cells in the center represent the freeway, and the outer and cross cells represent arterial streets. N_i^t for freeway cells and arterial cells are 20 and 10, respectively; Q_i^t for freeway cells and arterial cells are 20 and 10, respectively. The only destination in this network is cell 68. The demand levels originating from cells 1, 2 and 3 are 1,800, 3600 and 1800 vph, respectively. The demand is uniformly distributed over the planning period. The iteration limit is set to be 10 in this experiment.

In this network, we randomly generate five sets of cell capacities and apply the heuristic scheme to calibrate the DTA. The CPU time required to complete the

calibration of a network with this size is around 10 minutes. The results are summarized in FIGURE 10.

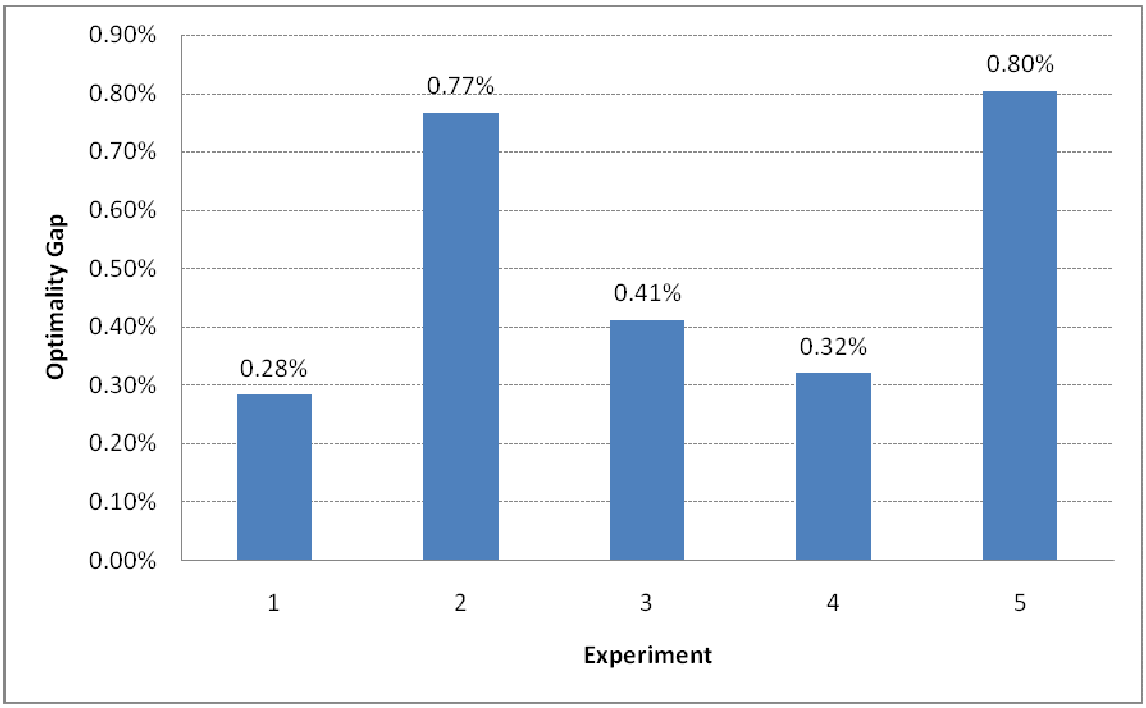

FIGURE 10: Numerical Results of the 68-cell CTM Network

The optimality gap shown in the figure is calculate by $\frac{\sum_{i \in C \setminus C_S} \sum_{t \in T} | x_i^t - x_{i,a}^t |}{\sum_{i \in C \setminus C_S} \sum_{t \in T} x_{i,a}^t}$.

Though the proposed heuristic does not match the actual occupancies perfectly, the maximum optimality gap is less than 1% in the tested five cases. The optimality gap could be the result of using approximated dual variables instead of exact ones or the heuristic nature of the whole decomposition scheme. It can be observed from FIGURE 10 that the performance of the proposed approach varies in different experiments. Inherently, certain problems are harder to solve than others. The traffic bottleneck due to the theoretical CTM may be the reason because downstream bottlenecks have a great

impact on upstream traffic. For instance, in a freeway link, a bottleneck can take place in downstream cells when the upstream cells have higher capacities. Thus, the randomly generated capacities can yield different bottleneck distributions in the network. Suppose that a bottleneck exists in the first cell of a link during the solution process and the count predicted by the UODTA is too low in the last cell of that link. The proposed approach may identify the first bottleneck cell and perturb its capacities accordingly. Though the perturbation may ease the congestion in the first cell and increase the flow between the cell and its downstream cell, it may create a bottleneck in the very next cell and direct the approach to perturb the capacities of that cell. This theoretical limit can potentially hamper the solution process and yield the performance differences observed in this numerical example. For details of the CTM theoretical limits, we refer to Karoonsoontawong (2006).

6.4 Summary

DTA models are increasingly employed by agencies and practitioners to address the unrealistic traffic representation assumptions commonly adopted in static traffic assignment. However, the calibration of the capacity values used in DTA models has typically been an onerous task without a systematic approach. In this chapter, we present a Dantzig-Wolfe decomposition-based heuristic that exploits the mathematical programming structure of the underlying cell transmission theory to calibrate the capacities of DTA in a computationally efficient manner. From the preliminary results, the proposed heuristic can accurately calibrate the network capacities and match the counts predicted by the DTA and the target counts within 1% of the optimality gap among all conducted experiments.

It should be noted that the proposed heuristic has the potential to scale. For instance, instead of calibrating the capacities at cell level, one can instead calibrate the

capacities at link level. The dual variable approximations can then be applied at the same level such that the computational efficiencies can be significantly improved.

Though this work presents a methodology that can calibrate the network capacities, it is not without its share of limitations. First, the proposed heuristic needs to be tested on an even larger network to demonstrate its efficacy in real-world applications. To accomplish this, a large-scale DTA simulation module may be incorporated to enhance the computational efficiency, and the scalability potential of the heuristic should be explored (this requires computational modifications well beyond the methodological development presented here). Secondly, some techniques used in the heuristic are designed specifically for a network with only a single destination (namely the analytical combinatorial algorithm for DTA). Though some researchers employed single destination models to analyze some specific applications, such as evacuation, multiple destination problems should be considered in future research either through multi-destination combinatorial methods or simulation-based DTA approaches.

Chapter 7. Single-destination Dynamic Congestion Pricing

The objective of dynamic congestion pricing is to determine time-varying tolls to maximize system performance when users route themselves to minimize their generalized cost in a greedy manner. In this chapter, we first introduce the mathematical formulation of the problem and present a heuristic scheme to tackle it. It is worth noting that we do not consider elastic demand in this dissertation. In other words, the time-varying demand considered in this dissertation is fixed and will not be affected by the tolls imposed. The detailed formulation is as follows.

Dynamic Congestion Pricing Formulation

$$\underset{x,y}{Min} \sum_{(i,j)\in E_S} \sum_{t\in T} \left(t \cdot y_{ij}^t\right) \tag{7.1}$$

subject to

$$\Psi(\Xi^*, \omega)'(\Xi - \Xi^*) \geq 0 \qquad \forall \Xi \in D \tag{7.2}$$

where

$$D = \left\{ (x,y): \begin{array}{lll}
x_i^t - x_i^{t-1} + \sum_{(i,j)\in FS(i)} y_{ij}^{t-1} - \sum_{(j,i)\in RS(i)} y_{ji}^{t-1} = d_i^t & \forall i \in C \setminus C_s, t \in T & (7.3.1) \\
\sum_{(i,j)\in FS(i)} y_{ij}^t - x_i^t \leq 0 & \forall i \in C \setminus C_s, t \in T & (7.3.2) \\
\sum_{(j,i)\in RS(i)} y_{ji}^t \leq \delta_i^t (N_i^t - x_i^t) & \forall i \in C \setminus C_s, t \in T & (7.3.3) \\
\sum_{(j,i)\in RS(i)} y_{ji}^t \leq Q_i^t & \forall i \in C \setminus C_s, t \in T & (7.3.4) \\
\sum_{(i,j)\in FS(i)} y_{ij}^t \leq Q_i^t & \forall i \in C \setminus C_s, t \in T & (7.3.5) \\
x_i^0 = \zeta_i & \forall i \in C \setminus C_S & (7.3.6) \\
y_{ij}^0 = 0 & \forall (i,j) \in E & (7.3.7) \\
x_i^{|T|} = 0 & \forall i \in C \setminus C_S & (7.3.8) \\
x_i^t \geq 0 & \forall i \in C \setminus C_S, t \in T & (7.3.9) \\
y_{ij}^t \geq 0 & \forall (i,j) \in E, t \in T & (7.3.10)
\end{array} \right\}$$

To the best of our knowledge, there is no apparent lower-level objective function that characterizes dynamic user equilibrium traffic assignments *with* tolls when the underlying traffic flows are based on the CTM (as mentioned earlier, the existing objective function only features UODTA *without* tolls). Therefore, the UODTA *with* tolls is framed as a variational inequality in generalized cost (Eq. (7.2)). The generalized cost corresponds to the sum of the tolls paid and the travel time experienced scaled up by the users Value of Time (VOT). It can be observed that the user equilibrium flows Ξ^* always result in lower generalized route costs than other feasible dynamic traffic assignments by rearranging Eq. (7.2) to $\Psi(\Xi^*, \omega)'\Xi \geq \Psi(\Xi^*, \omega)'\Xi^*$. The feasible region (*D*) includes the CTM related constraints, since we employ the CTM to represent traffic dynamics.

The above formulation essentially corresponds to a Mathematical Program with Equilibrium Constraints (MPEC) in which there is an objective function defined on a non-convex feasible region. Thus, there can be multiple local optima and traditional mathematical programming algorithms may not guarantee convergence to a global optimal solution. Even though there have been significant advances in developing solvers for MPECs, most of them are not efficient in solving problems of significant size. Therefore, we first introduce an exact approach in the next section, and then we propose a MSA-based heuristic in following section to obtain time-varying tolls in larger networks.

7.1 Solution Methodology: Exact Approach

It is worth noting that the widely used static congestion pricing approach may not apply directly in the dynamic case considered in this paper. The explanation is as follows. Denote τ as the private travel time faced by an entering road user and *v* as the number of trips. Thus total system cost equals $\tau \cdot v$, and the marginal social cost of an additional trip is $\frac{\partial \tau \cdot v}{\partial v} = \tau + \frac{\partial \tau}{\partial v} \cdot v$. Therefore, the value of $\frac{\partial \tau}{\partial v} \cdot v$ constitutes the

optimal toll in static marginal pricing scheme. In addition, there exists a convenient Bureau of Public Roads (BPR. 1964) function that can be used to calculate static toll prices. However, it is not straightforward to apply this approach in dynamic congestion pricing. Essentially, dynamic traffic assignment (either system-optimal or user-optimal) is a linear program, provided that the underlying traffic dynamics are characterized by CTM (Ziliaskopoulos (2000) and Ukkusuri (2002)). No practical link performance function is available except for the computationally expensive dynamic traffic simulation. In other words, it is not computationally cheap to evaluate $\frac{\partial \tau}{\partial v} \cdot v$ in a dynamic traffic assignment. Thus, fundamental problem features have to be investigated to develop an efficient approach.

It can be observed that the only difference between SODTA LP and UODTA LP presented in Chapter 3 is the parameters employed in the objective functions. The objective function of a SODTA is $\sum_{\forall (i,j) \in E_s} \sum_{\forall t \in T} t \cdot y_{ij}^t$ while the objective function of a UODTA is $\sum_{(i,j) \in E_S} \sum_{t \in T} \left(M_t \cdot y_{ij}^t \right)$. Thus, the difference between these two objective functions should be the time-varying tolls to be imposed. In other words, the following equation provides a convenient approach to determine the exact toll prices ϕ_i^t that shift UODTA user behaviors to SODTA ones:

$$\phi_i^t = t - M_t \qquad \forall i \in C_S, t \in T$$

The drawbacks of this approach are threefold: (1) the approach determines the toll prices solely in the sink cell, instead of cells in the candidate links; (2) the cost vector M_t grows exponentially, which makes this approach unsuitable for networks with a

large number of users; (3) the toll prices can be negative, which is not feasible in practice unless government agencies are willing to pay road users for using a certain roads. To overcome these drawbacks, we proposed a heuristic scheme. Details of the heuristic scheme are presented in the following section.

7.2 Solution Methodology: MSA-Based Heuristic

Due to the inefficiencies involved in solving the dynamic pricing problem using existing MPEC algorithms, a MSA-based heuristic has been developed in this section. In the heuristic, appropriate tolls are determined using dual variable approximation techniques and are averaged with MSA techniques across iterations. Then, the user equilibrium under tolls is evaluated using a combinatorial heuristic. The procedure is repeated until convergence. The next section provides an overview of dual variable approximation, where time-varying tolls are determined for a given set of flows using the dual variable approximation procedure.

7.2.1. Dual Variable Approximation Procedure

The fundamental logic behind the dual variable approximation procedure is that dual variables for the flow balance, capacity and jam density constraints in the bi-level formulation can be used as a proxy for dynamic toll prices. The interpretations of dual variables associated with CTM related constraints will be detailed in the later formulation (Eqs. (7.4)-(7.14)). Essentially, dual variables reflect the level of congestion in the network. If a cell is congested, the dual variable for that cell will not be zero. Moreover, the greater the congestion, the greater the value of the dual variables associated with that cell. Therefore, the time-varying toll prices may be inferred by extracting the dual variables of the above constraints.

However, no known technique exists that can obtain the exact dual variables for this specific bi-level program (Eqs. (7.1)-(7.3)). For specific values of user equilibrium flows, the corresponding dual variables can be approximated according to an upper-level objective function based on the single level formulation, as shown in Eqs. (7.4)-(7.14), where $\pi_i^{0,t}$ - $\pi_i^{4,t}$ in the formulation are the dual variables of the corresponding constraints. For readability, we present the complete formulation, though some of the equations are similar to those presented in earlier sections.

$$\underset{\phi,x,y}{Min} \sum_{(i,j)\in E_S} \sum_{t\in T} \left(t \cdot y_{ij}^t\right) \tag{7.4}$$

subject to

$$x_i^t - x_i^{t-1} + \sum_{(i,j)\in FS(i)} y_{ij}^{t-1} - \sum_{(j,i)\in RS(i)} y_{ji}^{t-1} = d_i^t \qquad \forall i \in C \setminus C_s, t \in T \qquad : \pi_i^{0,t} \tag{7.5}$$

$$\sum_{(i,j)\in FS(i)} y_{ij}^t - x_i^t \le 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{1,t} \tag{7.6}$$

$$\sum_{(j,i)\in RS(i)} y_{ji}^t \le \delta_i^t (N_i^t - x_i^t) \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{2,t} \tag{7.7}$$

$$\sum_{(j,i)\in RS(i)} y_{ji}^t \le Q_i^t \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{3,t} \tag{7.8}$$

$$\sum_{(i,j)\in FS(i)} y_{ij}^t \le Q_i^t \qquad \forall i \in C \setminus C_S, t \in T \qquad : \pi_i^{4,t} \tag{7.9}$$

$$x_i^0 = \zeta_i \qquad \forall i \in C \setminus C_S \tag{7.10}$$

$$y_{ij}^0 = 0 \qquad \forall (i,j) \in E \tag{7.11}$$

$$x_i^{|T|} = 0 \qquad \forall i \in C \setminus C_S \tag{7.12}$$

$$x_i^t \geq 0 \qquad \forall i \in C \setminus C_S, t \in T \qquad (7.13)$$

$$y_{ij}^t \geq 0 \qquad \forall (i, j) \in E, t \in T \qquad (7.14)$$

Effective congestion pricing scheme needs to prescribe fees that force users to internalize the externalities they incur to the system. In this paper, dual variables $\pi_i^{0,t} - \pi_i^{4,t}$ are chosen as the approximated externalities when an additional road user is added to the system. Note that the dual variable of Eq. (7.5) indicates the TSTT change due to the increase of demand in cell *i* at time *t*. Thus, the value of $\pi_i^{0,t}$ quantifies the private cost faced by the entering road user. However, there is not a computationally cheap approach to acquire the impact of that particular user on other road users. Thus, we choose to employ the value of $\pi_i^{0,t}$ as the approximated impact. In addition to the impact on the TSTT, the user as well consumes the capacities (saturation flow rate and jam density) when the user appears in cell *i* at time *t*. The impact of those capacity consumptions on the system can be quantified by dual variables $\pi_i^{1,t} - \pi_i^{4,t}$. Therefore, the overall externalities that should be imposed to cell *i* at time *t* is $\sum_{k=0}^{4} \pi_i^{k,t}$.

After solving the UODTA with tolls, the occupancies (*x*) and flow rates (*y*) of cells at different time intervals can be obtained. We then approximate the dual variables according to the dynamic occupancies and flow rates. The approximation techniques are described as follows.

7.2.2. Approximation of $\pi_i^{0,t}$

As mentioned above, the dual variable associated with Eq. (7.5) denotes the change in TSTT due to the change in OD demand in cell *i* at time interval *t*. Therefore,

to approximate the dual variables, a user is added to cell *i* at time interval *t*. The user then follows his/her time-dependent shortest path (TDSP) to the destination. We employ the travel time of the TDSP as the proxy of dual variables $\pi_i^{0,t}$.

To efficiently approximate the dual variable $\pi_i^{0,t}$, we employ the backward connectivity algorithm proposed earlier. The procedure first generates a time-expanded network $c(i,t)\ \forall i \in C, t \in T$ and scan eligible list $LIST$. In the time-expanded network, the sink cell is assigned a label ($Label(c(i,t)) \leftarrow T - t + 1,\ \forall i \in C_S, t \in T$) according to its time interval (*t*), and the labels of all other cells are set to zero ($Label(c(i,t)) \leftarrow 0,\ \forall i \in C \setminus C_S, t \in T$).

Initially, the algorithm places $c(i,t)\ \forall i \in C_S, t \in T$ into the scan eligible list $LIST$. The algorithm selects a cell $c(i,t)$ from $LIST$, and then a search procedure checks the connectivity from that cell to its upstream cells ($\Gamma^{-1}(c(i,t))$). If $Label(\Gamma^{-1}(c(i,t))) < Label(c(i,t))$, cell $\Gamma^{-1}(c(i,t))$ is not congested and the connector connecting the two cells is not saturated: $Label(\Gamma^{-1}(c(i,t))) \leftarrow Label(c(i,t))$ and $LIST = LIST \cup \{\Gamma^{-1}(c(i,t))\}$. The algorithm then selects the next cell from $LIST$ and repeats the procedure until $LIST$ is empty or a source cell is reached. The process is implemented in a dequeue data structure (Ahuja, Magnanti, and Orlin, 1993) to reduce the computational efforts in practice. When the algorithm terminates, the travel time of the TDSP ($TT(c(i,t))$) from cell *i* at time interval *t* to its destination is the difference between the final label and the index of the sink cell at this time interval. In other words,

$$TT(c(i,t)) \leftarrow Label(c(i,t)) - Label(c(j,t)), \quad \forall i \in C \setminus C_S, j \in C_S, t \in T.$$

The algorithm finds the travel times of TDSPs without explicitly finding the TDSPs and significantly improves the computational efficiency.

7.2.3. Approximation of $\pi_i^{1,t}$, $\pi_i^{2,t}$, $\pi_i^{3,t}$ *and* $\pi_i^{4,t}$

In this research, we adopt similar dual variable approximation techniques to those presented in chapter 4 and approximate the dual variables $\pi_i^{1,t}$, $\pi_i^{2,t}$, $\pi_i^{3,t}$ and $\pi_i^{4,t}$ by the complimentary slackness conditions. In this section, we review the approximation scheme without a detailed description:

I. If $\sum_{(i,j)\in FS(i)} y_{ij}^t - x_i^t < 0$, then set $\pi_i^{1,t} = 0 \quad \forall i \in C \setminus C_S, t \in T$.

II. If $\sum_{(j,i)\in RS(i)} y_{ji}^t < \delta_i^t (N_i^t - x_i^t)$, set $\pi^{2,t} = 0 \quad \forall i \in C \setminus C_S, t \in T$.

III. If $\sum_{(j,i)\in RS(i)} y_{ji}^t < Q_i^t$, set $\pi_i^{3,t} = 0 \quad \forall i \in C \setminus C_S, t \in T$.

IV. If $\sum_{(i,j)\in FS(i)} y_{ij}^t < Q_i^t$, set $\pi_i^{4,t} = 0 \quad \forall i \in C \setminus C_s, t \in T$.

V. If $x_i^t > 0$, set $\pi_i^{1,t} = \pi_i^{0,t+1} - \pi_i^{0,t}$. Else if $x_i^t = 0$, we approximate $\pi_i^{1,t}$ by the equality $\pi_i^{1,t} = \pi_i^{0,t+1} - \pi_i^{0,t}$.

VI. If $x_i^t > 0$, set $\pi_i^{1,t} - \delta_i^t \pi_i^{2,t} = \pi_i^{0,t+1} - \pi_i^{0,t}$. Else if $x_i^t = 0$, we approximate the dual variables by treating the inequality $\pi_i^{1,t} - \delta_i^t \pi_i^{2,t} \le \pi_i^{0,t+1} - \pi_i^{0,t}$ as equality. If $\pi_i^{1,t}$ and π_i^{2t} are both undetermined, we assume $\pi_i^{1,t} = 0$ and $\pi_i^{2t} = -(\pi_i^{0,t+1} - \pi_i^{0,t})$.

VII. If $y_{ij}^t > 0$, $\pi_j^{3,t} + \pi_i^{4,t} = \pi_i^{0,t+1} - \pi_j^{0,t+1} - \pi_i^{1,t} - \pi_j^{2,t}$. Else if $y_{ij}^t = 0$, we approximate the dual variables by treating the inequality $\pi_j^{3,t} + \pi_i^{4,t} \geq \pi_i^{0,t+1} - \pi_j^{0,t+1} - \pi_i^{1,t} - \pi_j^{2,t}$ as equality. If $\pi_j^{3,t}$ and π_i^{4t} are both undetermined, we assume $\pi_j^{3,t} = \pi_i^{4t}$.

VIII. If $y_{ij}^t > 0$, $\pi_i^{4,t} = \pi_i^{0,t+1} - \pi_i^{1,t} - t$. Else if $y_{ij}^t = 0$, we approximate the dual variables by treating the inequality $\pi_i^{4,t} \geq \pi_i^{0,t+1} - \pi_i^{1,t} - t$ as equality

Given a set of flows, once the dual variables are approximated, the new toll prices are determined by averaging the dual variables using the MSA procedure (Eq. 7.15) and then summing them (Eq. 7.16). It should be noted that the heuristic may oscillate between solutions and fail to converge without the MSA procedure.

$$\pi_i^{k,t,iter} \leftarrow \frac{\pi_i^{k,t,iter}}{iter} + \frac{\pi_i^{k,t,iter-1} \times (iter-1)}{iter} \quad \forall k \in \{0,1,2,3,4\} \tag{7.15}$$

$$\phi_i^t \leftarrow \sum_{k=0}^{4} \pi_i^{k,t,iter} \quad \forall i \in C \setminus C_S, t \in T. \tag{7.16}$$

Once these prices are imposed, the user equilibrium flows under tolls is calculated using the combinatorial procedure proposed in the next section.

7.3 Combinatorial Heuristic for UODTA with Tolls

The purpose of the combinatorial heuristic is to determine the user equilibrium flows when time varying tolls are levied on the network. The heuristic works on the principle of incrementally assigning demand onto paths in a time-expanded CTM network and reducing the capacity of the cells along the assigned path.

The combinatorial heuristic can be explained with the help of an example cell network. Suppose that we have a 6-cell CTM network (FIGURE 2) and we generate the time-expanded CTM network depicted in FIGURE 11. Let the numbers in the cells be the exogenous tolls decided and imposed by other procedures. For simplicity, we assume that the user's value of time λ is \$1 per time unit. A user departs from cell 1 at the fixed departure time 1 and tries to arrive at the destination cell 6.

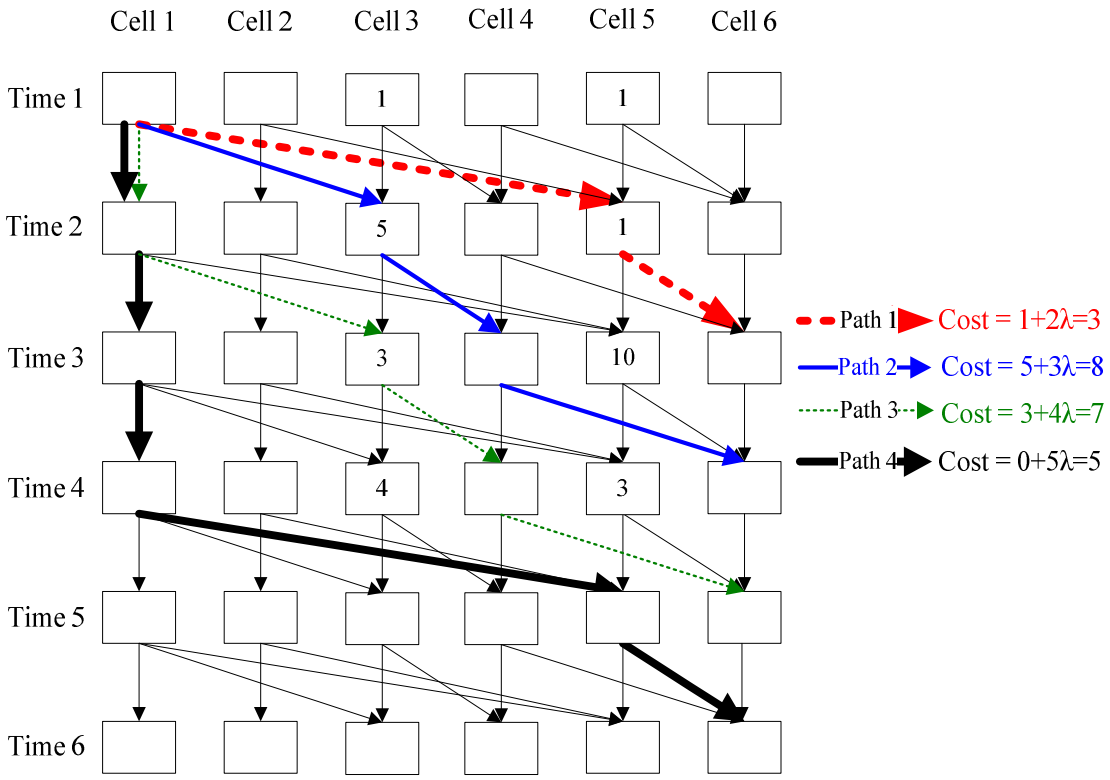

FIGURE 11: UODTA with Tolls

The heuristic first finds the shortest paths in terms of generalized cost from cell 1 to cell 6 at different time intervals until (1) a path with zero cost is found or (2) the path to cell 6 at the end of planning horizon time 6 is found. In the example above, four paths can be determined and the associated generalized path costs are shown in FIGURE 11. For instance, the user will have to pay the toll cost (\$1) and time cost

($2 $(=2\times\lambda)$) if the user decides to take path 1. The user essentially faces these four choices. Under the rational assumption, the user should choose path 1 instead of other three paths since it has the lowest generalized path cost. The jam densities and saturation flow rates along path 1 are then reduced by one unit accordingly. After assigning the user, the heuristic finds the path for the next user based on the same reasoning until all users are assigned their paths. When all the users have been assigned there is no incentive for any user to unilaterally shift paths to improve their travel cost. The procedure of UODTA with the exogenous tolls is completed.

Note that the departure time decision is implicitly considered in the heuristic. For instance, if the user chooses to follow path 4, the user essentially decides to stay in cell 1 for three time intervals and depart at time interval 4 instead of time 1.

7.4 MSA-based Heuristic Overview

This section provides a summary of the MSA-based heuristic to determine the dynamic prices. To be precise, the pseudo code of the heuristic is presented as follows:

Pseudo code of the MSA-based Heuristic

Data: $N_i^t, Q_i^t, d_i^t, \delta_i^t \quad \forall i \in C, t \in T$, *MAX_ITERATION*

Step 0: Set *iter* = 1, $\phi_i^t = 0 \quad \forall i \in C \setminus C_S, t \in T$

Step 1: Solve the UODTA with tolls $\phi_i^t \quad \forall i \in C \setminus C_S, t \in T$ by heuristics proposed in the pervious section to obtain $x_i^t \ \forall i \in C \setminus C_S, t \in T$, $y_{ij}^t \ \forall (i, j) \in E, t \in T$ and TSTT.

Step 2: Approximate dual variables $\pi_i^{0,t,iter}, \pi_i^{1,t,iter}, \pi_i^{2,t,iter}, \pi_i^{3,t,iter}, \pi_i^{4,t,iter}$ based on the solution in Step 1

Step 3: If no user can unilaterally shift paths to improve their travel cost, stop and report tolls $\phi_i^t \quad \forall i \in C \setminus C_S, t \in T$. Otherwise, go to Step 4.

Step 4: Use the Method of Successive Average (MSA)-type equation to average the approximated dual variables. That is,

$$\pi_i^{k,t,iter} \leftarrow \frac{\pi_i^{k,t,iter}}{iter} + \frac{\pi_i^{k,t,iter-1} \times (iter-1)}{iter} \qquad \forall k \in \{0,1,2,3,4\}$$

Step 5: Set $\phi_i^t \leftarrow \sum_{k=0}^{4} \pi_i^{k,t,iter} \quad \forall i \in C \setminus C_S, t \in T$.

Step 6: Set *iter* = *iter* +1.

If *iter* > *MAX_ITERATION*, stop and report tolls $\phi_i^t \quad \forall i \in C \setminus C_S, t \in T$.

Otherwise, go to Step 1.

The input data of the heuristic includes the time-dependent jam density N_i^t, saturation flow rates Q_i^t, O-D demand d_i^t, shockwave parameter δ_i^t and number of maximum iterations. In Step 0, the iteration counter and tolls are initialized with 1 and 0's, respectively. UODTA with tolls is then solved in Step 1. To determine the time-varying tolls, the dual variable approximation techniques are employed in Step 2. If the stopping criterion is met in Step 3, the heuristic stops. Otherwise, it continues to average the tolls with MSA techniques in Step 4, set the new tolls in Step 5, increment the counter in Step 6 and repeat the procedure. Note that exact dynamic tolls can shift UODTA to SODTA in one iteration and eliminate the need for the iterative procedure. However, due to the heuristic nature of the dual approximation techniques and combinatorial UODTA heuristic, iterative process combining MSA is necessary for better convergence.

Even though a rigorous proof of convergence is not provided, the heuristic converges for most cases because the dual variables correspond to the descent direction or the negative gradient with respect to the upper-level objective function. This procedure is similar to the sensitivity analysis based procedure for calculating gradients of static traffic equilibrium solutions first introduced by Tobin and Friesz (1988), which have been used to solve a wide variety of bi-level programs involving network design,

tolling and origin-destination matrix estimation. A detailed review of some of the applications of this method can be found in Jossefsson and Patrikson (2007). The method presented in this paper is conceptually similar in the sense that, for the proposed problem, the gradients correspond to the dual variables. This is primarily made possible due to the linear programming structure of the dynamic assignment problem. However, this heuristic is unique because it is the first time such a gradient approximation technique has been applied to determine time-dependent tolls.

7.5 Numerical Experiments

Based on the suggested VOT by the Oregon Department of Transportation (2003) the average VOT for automobile drivers is assumed to be $15.31 /hour. We employ this homogeneous VOT for all users in this dissertation. The 6-cell CTM network contains 6 cells and 6 cell connectors, as depicted in FIGURE 2. There are two source cells (cell 1 and cell 2) and one sink cell (cell 6) in the network.

The time-varying characteristics of the network (jam density N_i^t, saturation flow rate Q_i^t) are described in TABLE 17. The planning period is eight time intervals with two OD pairs. The seven time-dependent OD demands are given in TABLE 18.

TABLE 17: Characteristics of 6-cell CTM Network

Cell	N_i^t	Q_i^t
1	+inf	1
2	+inf	1
3	2	1
4	2	1
5	2	1
6	+inf	+inf

TABLE 18: Time-dependent OD Demands for 6-cell CTM Network

	Origin	Destination	Demand
Time 1	1	6	1
	2	6	1
Time 2	1	6	1
	1	6	1
	2	6	1
	2	6	1
Time 3	2	6	1

7.5.1. First-Best Pricing

In the 6-cell CTM network, we experiment with both the first-best pricing and second-best pricing. In the first-best pricing, tolls are allowed to be levied in all cells in the network. Thus, we apply the dual variable approximation techniques to determine the time-varying tolls for all cells. To compare the solution obtained by the heuristic, we solve the SODTA with CPLEX using the SODTA LP formulation proposed by Ziliaskopoulos (2000). The numerical results, together with the SODTA, are summarized in TABLE 19.

TABLE 19: First-Best Pricing

SODTA: path (*time*)	**UODTA without Tolls: path (*time*)**	**UODTA with Tolls: path (*time*)**
1-3-4-6 (*1-2-3-4*)	**1-5-6** (*1-2-3*)	**1-3-4-6** (*1-2-3-4*)
2-5-6 (*1-2-3*)	**2-2-5-6** (*1-2-3-4*)	**2-5-6** (*1-2-3*)
1-3-4-6 (*2-3-4-5*)	**1-1-5-6** (*2-3-4-5*)	**1-3-4-6** (*2-3-4-5*)
1-1-3-4-6 (*2-3-4-5-6*)	**2-2-2-5-6** (*2-3-4-5-6*)	**1-1-3-4-6** (*2-3-4-5-6*)
2-5-6 (*2-3-4*)	**2-2-2-5-6** (*3-4-5-6-7*)	**2-5-6** (*2-3-4*)
2-2-5-6 (*2-3-4-5*)	**1-3-4-6** (*2-3-4-5*)	**2-5-6** (*3-4-5*)
2-2-5-6 (*3-4-5-6*)	**2-2-2-2-2-5-6** (*2-3-4-5-6-7-8*)	**2-2-2-5-6** (*2-3-4-5-6*)
TSTT = 20	TSTT = 25	TSTT = 20

As can be seen from the table, without imposing the tolls in this network, the UODTA results in the total system travel time of 25. However, the total system travel time decreases to 20 with the tolls determined by the proposed procedure. With slightly different user paths, the MSA-based heuristic obtains the equivalent SODTA solution. The optimality gap of the proposed heuristic is 0% in this network.

7.5.2. Second-Best Pricing

Next, we test the second-best pricing by not allowing tolls in different cell across the planning horizon. As can be seen from the summarized results in TABLE 20, if cell 5 is not allowed to be tolled, the resulting TSTT is 5 units worse than when it is allowed. Users tend to choose the toll-free cell 5 in that scenario. Essentially, the user behaviors shift back to UODTA without tolls. Checking the UODTA without tolls solution presented in TABLE 19, we can see that cell 5 is more appealing to road users (6 out of 7 users choose to pass cell 5 in that case) and congestion in cell 5 then follows. Therefore, imposing tolls in cell 5 can lead to better system performance by shifting users to alternative routes (cell 1-cell 3-cell 4-cell 6). On the other hand, imposing tolls in cells 1-4 does not lead to any system improvement. This again proves that the approximated dual variables accurately locate the congestion, which means that the dual variables can serve as the descent direction when devising congestion reduction measures.

TABLE 20: Second-Best Pricing

	Cell that are not allowed to be tolled				
	Cell 1	**Cell 2**	**Cell 3**	**Cell 4**	**Cell 5**
Path (*time*)	**1-3-4-6** (*1-2-3-4*)	**1-3-4-6** (*1-2-3-4*)	**1-3-4-6** (*1-2-3-4*)	**1-3-4-6** (*1-2-3-4*)	**1-5-6** (*1-2-3*)
	2-5-6 (*1-2-3*)	**2-5-6** (*1-2-3*)	**2-5-6** (*1-2-3*)	**2-5-6** (*1-2-3*)	**2-2-5-6** (*1-2-3-4*)
	1-3-4-6 (*2-3-4-5*)	**1-3-4-6** (*2-3-4-5*)	**1-3-4-6** (*2-3-4-5*)	**1-3-4-6** (*2-3-4-5*)	**1-1-5-6** (*2-3-4-5*)
	2-5-6 (*2-3-4*)	**2-5-6** (*2-3-4*)	**2-5-6** (*2-3-4*)	**2-5-6** (*2-3-4*)	**2-2-2-5-6** (*2-3-4-5-6*)
	2-5-6 (*3-4-5*)	**2-5-6** (*3-4-5*)	**2-5-6** (*3-4-5*)	**2-5-6** (*3-4-5*)	**1-3-4-6** (*2-3-4-5*)
	1-1-3-4-6 (*2-3-4-5-6*)	**1-1-3-4-6** (*2-3-4-5-6*)	**1-1-3-4-6** (*2-3-4-5-6*)	**1-1-3-4-6** (*2-3-4-5-6*)	**2-2-2-5-6** (*3-4-5-6-7*)
	2-2-2-5-6 (*2-3-4-5-6*)	**2-2-2-5-6** (*2-3-4-5-6*)	**2-2-2-5-6** (*2-3-4-5-6*)	**2-2-2-5-6** (*2-3-4-5-6*)	**2-2-2-2-2-5-6** (*2-3-4-5-6-7-8*)
TSTT	20	20	20	20	25

Next, we conduct the sensitivity analysis of the VOT with the first-best pricing scenario. Note that we do not consider elastic demand in this dissertation. Thus, the total number of demand is fixed in all the scenarios. It can be observed from the summarized results in TABLE 21 that when the VOT is higher ($4\times\lambda$ and $5\times\lambda$), users are less sensitive to tolls imposed. In other words, users tend to choose routes with less travel time instead of choosing the route with lower toll cost. This causes worse system performance in these two cases. On the other hand, when VOT is lower, the users are relatively more sensitive to tolls. Thus, better total system performance can be achieved with effective tolls.

TABLE 21: First-best Pricing with Different VOT ($\lambda = \$15.31/hour$)

VOT ($/hour)	TSTT
λ	20
$2\times\lambda$	20
$3\times\lambda$	20
$4\times\lambda$	22
$5\times\lambda$	22

7.5.3. 68-cell CTM Network

Next, we conduct the experiments on a larger network to further investigate the performance of the proposed heuristic. The 68-cell CTM network (depicted in FIGURE 3) consists of 68 cells and 74 cell connectors, including three source cells (cell 1, 2 and 3) and one sink cell (cell 68). The cells in the center represent the freeway and the outer and cross cells represent arterial streets. The deterministic time-varying O-D demand and time-dependent network characteristics are described in

TABLE 4 and TABLE 5.

To obtain the comparable solution, we again solved the SODTA with CPLEX. For network of this size, the LP formulation is composed of 49,982 constraints and 21,443 variables. The number of constraints and variables grow significantly with the network size. Algorithm that is capable of dealing with large-scale LPs may be necessary for even larger network (Li et al., 2003). Note that it is computational difficult to employ exact algorithm to solve the bi-level formulation or the VI based congestion pricing problem. With the combinatorial nature of the dual variable approximation and the MSA procedures, the proposed heuristic in fact has the potential to scale. The computational results of this network are summarized in TABLE 22.

Denote *H* as the solution obtained from the proposed heuristic, *S* as the result of SODTA and *U* as the result of UODTA. The maximum optimality gap $\left(\frac{H-S}{S}\right)$ of the proposed heuristic is 2.03% with different demand levels. It should be noted that the proposed heuristic does not allow fractional flows, while the SODTA LP formulation does. The fractional flow issues can potentially lead to a gap in the numerical solutions. In the last column of TABLE 22, we also present the heuristic's maximum possible percent gain, which is *(U-S)*.

TABLE 22: TSTT for Different OD Demand in the 68-cell CTM Network

Demand (D)	Heuristics (H)	SODTA (S)	UODTA (U)	Convergence Iteration	$\left(\frac{H-S}{S}\right)$	$\left(\frac{H-S}{U-S}\right)$
1.0*D	7,956	7,812	7,956	58	1.84 %	100.00 %
1.5*D	12,439	12,382	15,782	71	0.46 %	1.69 %
2.0*D	23,858	23,384	27,716	95	2.03 %	10.94 %
2.5*D	37,797	37,391	42,317	*	1.09 %	8.25 %
3.0*D	62,370	62,024	69,177	*	0.56 %	4.84 %

*Fail to converge before reaching iteration limit 100

With the original demand level, the heuristic obtains the solution with 100% maximum possible gain. However, the obtained gain is lowered as the demand level increases. One possible reason is that the imposed tolls have less impact, since the network tends to be over-congested in those cases and users cannot freely switch routes.

7.6 Summary

Congestion pricing has increasingly been seen as a powerful tool for both managing congestion and generating revenue for infrastructure maintenance and development. This chapter contributes to the growing body of literature in congestion pricing by providing the mathematical formulations and an efficient MSA-based heuristic

to determine time-varying tolls on networks. Extensive numerical experiments have been conducted on networks with various sizes to show the effectiveness and efficiency of the proposed heuristic. From the preliminary results, the heuristic finds solutions with the maximum optimality gap of 2.03% among all cases.

One advantage of the proposed heuristic is that users' decision on departure time is implicitly considered without extra work. Besides, the second-best pricing can be easily incorporated into the solution procedure by simply skipping the dual variable approximation procedure in predefined cells at specified time intervals. From the computational point of view, the proposed heuristic even benefits from second-best pricing due to the fewer dual variable approximation procedures required. In addition, user heterogeneity can be incorporated into the solution procedures by using the different VOT value for different users.

Though the overall heuristic provide a general framework that can efficiently solve dynamic congestion pricing problem with reasonable optimality gaps, the dual variable approximation procedures obtain merely the proxy of dual variables in single-destination networks. Exploring the multiple-destinations problem and finding the exact dual variables are the potential future extension of this work.

Secondly, the time resolution required by CTM and dynamic congestion pricing should be different in practice. For instance, the time interval in CTM may be in seconds while realistic dynamic congestion pricing can be in minutes or even in hours. Though the proposed model and solution scheme do not explicitly consider the time resolution issue, they can serve as the first step toward achieving this refinement.

Another possible future extension is the elastic user demand when tolls are levied on the network. The inclusion of elastic demand in the dynamic context can increase the computational complexity dramatically though.

Chapter 8. Multiple-destination Bi-level Linear Programming Network Design Problem: A Quantum-Inspired Genetic Algorithm

Data stored in a classic computer is represented by patterns of grouped bits to represent numbers. A quantum computer, however, utilizes the qubit to conduct these tasks. One of the main differences is that each qubit can hold a value of one, zero or any superposition of the two numbers at the atomic level. The state of a single qubit can be represented as $\alpha|0\rangle+\beta|1\rangle$, while the probability of the system being in the state of zero is $|\alpha|^2$ and the probability of the system being in state one is $|\beta|^2$. Therefore, $|\alpha|^2+|\beta|^2=1$ must be satisfied. This equation, $|\alpha|^2+|\beta|^2=1$, is referred to as the *normalization of the state equation*. We use the following example to further explain quantum logic. Suppose that we have a two qubit system with two pairs of complex-numbered amplitudes:

$$\begin{bmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{3} \\ \frac{\sqrt{3}}{2} & \frac{2\sqrt{2}}{3} \end{bmatrix}$$

The four possible states of the system are $|00\rangle,|01\rangle,|10\rangle,|11\rangle$, and the system can be represented as:

$$\frac{1}{2}\times\frac{1}{3}|00\rangle+\frac{1}{2}\times\frac{2\sqrt{2}}{3}|01\rangle+\frac{\sqrt{3}}{2}\times\frac{1}{3}|10\rangle+\frac{\sqrt{3}}{2}\times\frac{2\sqrt{2}}{3}|11\rangle$$
$$=\frac{1}{6}|00\rangle+\frac{\sqrt{2}}{3}|01\rangle+\frac{\sqrt{3}}{6}|10\rangle+\frac{\sqrt{6}}{3}|11\rangle$$

It then can be observed that the probabilities of the system being in the state $|00\rangle,|01\rangle,|10\rangle$ and $|11\rangle$ are $\left(\frac{1}{6}\right)^2,\left(\frac{\sqrt{2}}{3}\right)^2,\left(\frac{\sqrt{3}}{6}\right)^2$ and $\left(\frac{\sqrt{6}}{3}\right)^2$ respectively. It is

straightforward that a quantum computer with n qubits can potentially represent 2^n different states simultaneously. This feature facilitates the design of efficient polynomial-time algorithms. In the next section, we detail the design of a quantum-inspired genetic algorithm (QGA) using the features discussed above.

8.1 Algorithmic Design

The QGA is modified from the algorithm presented by Han and Kim (2002). The intuition behind this procedure is that the value of a qubit in a chromosome that has worse fitness value should have a higher probability of moving toward the value of the corresponding qubit in the incumbent solution. On the other hand, if the current chromosome yields a better fitness value, the qubits within should have higher probability of remaining at their current status. Assume that a qubit in the chromosome under consideration has the value of 1. In addition, the chromosome corresponds to a worse fitness value than the incumbent solution, which has a value of 0 in the corresponding qubit. Intuitively, one would hope that the qubit in the chromosome under consideration has a higher probability of moving toward 0 in the next generation. However, if the chromosome under consideration has a better fitness value than the incumbent chromosome, keeping the value of 1 for this qubit would be an ideal choice. Applying this concept for each qubit in each chromosome, QGA is capable of improving solutions over generations. There are four basic steps in the proposed QGA. We present the pseudo code of the algorithm followed by detailed explanations.

Quantum-inspired Genetic Algorithm Pseudo code

Step 0: Parameters

Maximum number of generation: G
Population Number: n
Number of qubits: m

Total available budget: φ

Number of cells: c

Number of bits to represent one b_i^g : NB

Intermediate binary string $y = [x_1, x_2, ..., x_m]$, $x_i \in \{0,1\}, \forall i \in \{1,2,...,m\}$

Step 1: Initialization

Generation counter $g \leftarrow 1$

Incumbent $TSTT^* \leftarrow \infty$

Generate population $Q(g) = \{q_1^g, q_2^g, ..., q_n^g\}$

Individual qubit chromosome $q_i^g = \begin{bmatrix} \alpha_1^g & \alpha_2^g & ... & \alpha_m^g \\ \beta_1^g & \beta_2^g & ... & \beta_m^g \end{bmatrix}$,

$$\alpha_i^g \leftarrow \frac{1}{\sqrt{2}} \qquad \forall i \in \{1,2,...,m\}$$

$$\beta_i^g \leftarrow \frac{1}{\sqrt{2}} \qquad \forall i \in \{1,2,...,m\}$$

Step 2: Interpret $Q(g)$

For i = 1 to n

(1) Generate binary string $y = [x_1, x_2, ..., x_m]$ from $q_i^g = \begin{bmatrix} \alpha_1^g & \alpha_2^g & ... & \alpha_m^g \\ \beta_1^g & \beta_2^g & ... & \beta_m^g \end{bmatrix}$

For j= 1 to m

Generate a random number γ between 0 and 1

If $\gamma > \left|\alpha_j^g\right|^2$, then $x_j^g = 1$

Else $x_j^g = 0$

End for

(2) Interpret the binary string $y = [x_1, x_2, ..., x_m]$ to floating-point encoded budget

$b_i^g = [b_1, b_2, ..., b_c]$

$k \leftarrow 0$

For j= 1 to m

If j % NB = 0, then

$$b_k \leftarrow \frac{\left(\sum_{t=0}^{NB} 2^t \times x_t\right) \times TAB}{2^{NB} - 1}$$

$++k$

End if

End for

If $\sum_{k=1}^{c} b_k > TAB$

For j= 1 to c

$$b_j = b_j \times \frac{TAB}{\sum_{k=1}^{c} b_k} \quad \text{(Repair mechanism)}$$

End for

End if

End for

Step 3: Evaluate *Q(g)*

For *i*= 1 to *n*

Modify network capacity according to the budget allocation b_i^g

Run DTA with the modified network capacity to obtain $TSTT_i$

If $TSTT_i < TSTT^*$

$TSTT^* = TSTT_i$

$b^* = b_i^g$

End if

End for

Step 4: Update *Q(g)*

For *i*= 1 to *n*

For *j*= 1 to *m*

$$\begin{bmatrix} \alpha_i^{g+1} \\ \beta_i^{g+1} \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix} \begin{bmatrix} \alpha_i^{g} \\ \beta_i^{g} \end{bmatrix}$$

End for

End for

Step 5: Check Stopping Criterion

If ($g > G$)

report $TSTT^*$ and b^*

Else

$g \leftarrow g + 1$

Go to Step 2

Step 0 provides the parameters used in the subsequent steps. Each individual qubit chromosome is initialized with a series of $\frac{1}{\sqrt{2}}$ in Step 1 to specify that each qubit has a probability of 50% of being in the state of zero and 50% in the state of one. In Step 2, we map each q_i^g to an intermediate binary string y and then translate y into a floating-point encoded budget b_i^g. For each qubit *j* in the chromosome of generation *g*, we generate a random number γ between 0 and 1. If $\gamma > \left|\alpha_j^g\right|^2$, then x_j^g takes the

value of 1, otherwise, it takes the value of 0. This process repeats until the binary string *y* completes. Then, binary string *y* is translated into the budget allocation using the widely used procedure presented in the pseudo code. It is important to note that we address the floating point issue by applying multiple bits for each b_i in the translation step. The precision can be determined by the equation $2^{m-1} < \left((UB - LB) \times 10^{precision} + 1\right) \leq 2^m$. In a binary string used in this paper, the lower bound (LB) is 0 and upper bound (UB) is 1. We choose 14 bits (m=14) to represent one b_i. The corresponding precision is 4, which is sufficient for the problem considered. In the process, we need the additional repair mechanism demonstrated in Step 2 when the budget interpreted from q_i^g is greater than the total available budget ($\sum_{k=1}^{c} b_k > TAB$). Given the budget allocation policy from Step 2, Step 3 evaluates each q_i^g by conducting user optimal dynamic traffic assignment with the modified network capacities. The system performance (total system travel time) is obtained in this step and is later used as the performance measurement of the qubit chromosome. Virtually any analytical or simulation-based approach can be embedded in this step to do the functional evaluation. In Step 4, *Q(g)* is updated using the Rotation Gate (RG) developed by Han and Kim (2002). The RG is important in QGA since it represents the direction of convergence. Details of the RG are left to the next section. We use the maximum number of generations as the stopping criterion, as shown in Step 5.

8.2 Direction of Convergence: Rotation Gate

It can be observed that the QGA becomes a random search if no proper mechanism is devised to update and improve qubit chromosomes. The Rotation Gate (RG) is designed to update the qubit chromosome and direct the algorithmic convergence.

It should be emphasized that the RG mainly works on the intermediate binary string $y=[x_1,x_2,...,x_m]$ instead of the floating-point encoded budget allocation $b_i^g=[b_1,b_2,...,b_c]$. In this paper, we use the RG developed in Han and Kim (2002). The details of the RG are presented in TABLE 23.

TABLE 23: $\Delta\theta_i$ for Rotation Gate

Qubit x_i in current solution y	Qubit x_i^* in incumbent solution y^*	Current Solution ≥ Incumbent Solution ($TSTT(y) \geq TSTT(y^*)$)	$\Delta\theta_i$
0	0	false	0
0	0	true	0
0	1	false	0.01π
0	1	true	0
1	0	false	-0.01π
1	0	true	0
1	1	false	0
1	1	true	0

The first column in TABLE 23 represents the qubit x_i in the current binary string *y*, while the second column represents the corresponding qubit in the incumbent binary string y^*. The third column gives the logic check of whether $TSTT(y) \geq TSTT(y^*)$, and the fourth column decides the angles used based on the first three columns.

We next use an example to demonstrate how $\Delta\theta_i$ is determined and how it improves solutions. Suppose that we are updating a qubit chromosome with $x_i = 0$, $x_i^* = 1$ and $TSTT(y) < TSTT(y^*)$. From TABLE 23, $\Delta\theta_i$ should be 0.01π (row marked grey in the table) and the RG will be:

$$\begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix} = \begin{bmatrix} 0.999507 & -0.031395 \\ 0.031395 & 0.999507 \end{bmatrix}$$

Then α , β for this qubit in the next generation will be:

$$\begin{bmatrix} \alpha_i^{g+1} \\ \beta_i^{g+1} \end{bmatrix} = \begin{bmatrix} 0.999507 & -0.031395 \\ 0.031395 & 0.999507 \end{bmatrix} \begin{bmatrix} \alpha_i^{g} \\ \beta_i^{g} \end{bmatrix}$$

$$= 0.999507\alpha_i^g - 0.031395\beta_i^g + 0.031395\alpha_i^g + 0.999507\beta_i^g$$

Interpreting these numbers, we can see that this qubit has a higher probability of being in state zero (increased by 0.031395^2) and lower probability of being in state one (decreased by 0.031395^2) in the next generation. The probability changes indicate the potential improvement direction. For every qubit in each qubit chromosome, the same RG is applied during the update step. In the RG, $\Delta\theta_i$ determines the direction of convergence and needs to be calibrated. An improper $\Delta\theta_i$ can result in divergent QGA. After our verification, $\Delta\theta_i$ from Han and Kim (2002) gives a reasonable convergence direction. However, the angles have not been fully calibrated in this paper due to the prohibitive computation requirements. Rather, we present the sensitivity analysis of the angles to indicate the possibility of further improvement in the computational experiment section.

8.3 Computational Experiences

In this section, we demonstrate the effectiveness of the QGA through empirical studies on two networks of different sizes: Sioux Fall and Monticello. All programs are implemented in the standard ANSI C language. The numerical experiments are conducted on a Linux machine with an Intel 3.00GHz CPU and 32 GB memory. The DTA module VISTA (Waller and Ziliaskopoulos, 1998) is used to evaluate the budget allocation policies determined by QGA. Note that the QGA developed is general in nature and can incorporate virtually any DTA module to do the functional evaluation.

To obtain a comparable solution to the QGA, we employ the conventional genetic algorithm (GA) with a parameter crossover rate (0.6) and mutation rate (0.01). Single

point crossover and roulette-wheel selection are adopted as the genetic operator. Among the parameters we tested, GA performs the best with the above mentioned parameters. The capacity expansion parameters χ_i and ϕ_i are from Karoonsoontawong and Waller (2006), and the values of these two parameters are 0.03 and 0.05, respectively. Two networks and their numerical results are presented in the following sections.

8.3.1. Sioux Fall Network

The Sioux Fall network (depicted in FIGURE 12) includes 48 nodes and 124 links. After breaking the network into cells according to the cell transmission model, the network consists of 1,252 cells, which are all considered for capacity expansion. There are 33 Origin-Destination (O-D) pairs and the total number of O-D demands is 22,374 vehicles. As mentioned previously, we adopt the simulation-based DTA module VISTA as the evaluation function and the simulation duration is 3,600 seconds for this network.

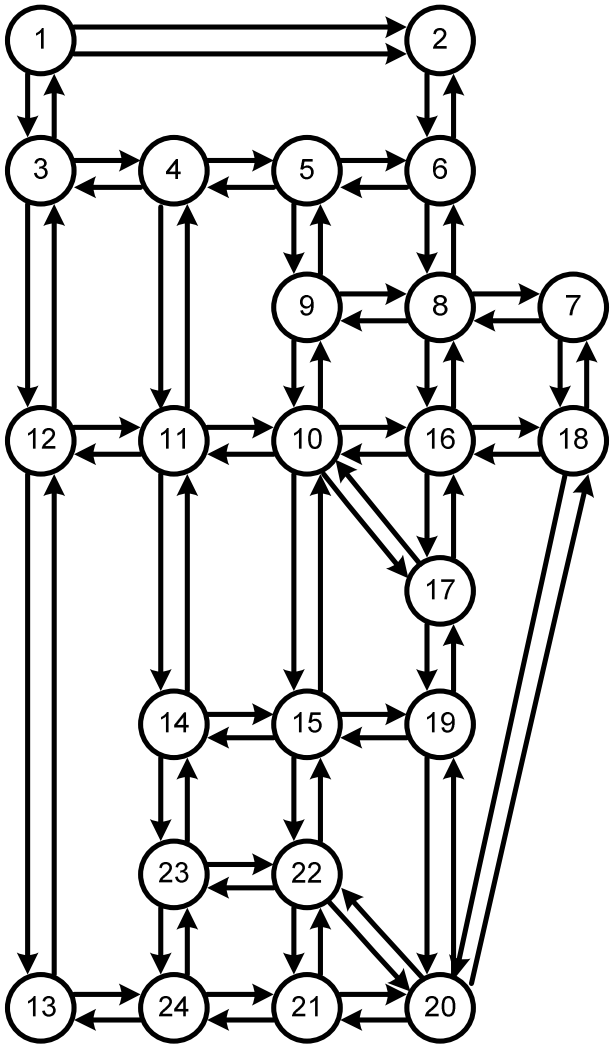

FIGURE 12: Sioux Fall Network

Note that it is computationally expensive to evaluate the budget allocation policies with the DTA module. From our preliminary experiments, it takes about 2 minutes to complete one DTA simulation using the traffic simulator VISTA. If we solve the dynamic network problem using QGA/GA with a population size of 100 and maximum generation of 100, it requires 10,000 functional evaluations in total. In other words, the computational time can easily reach 20,000 minutes even with a problem of such small size. Therefore, to demonstrate the solution quality and fast convergence of QGA

compared to GA, we limit the population size to 10 and the maximum number of generations to 10 to obtain solutions within reasonable computational time limits. With these predefined parameters, the computational time required for 100 functional evaluations (DTA simulation) is about 200 minutes for a single run, depending on the system performance when conducting the experiment. Due to the probabilistic nature of QGA, we average over 10 runs in all the experiments presented in this paper to obtain the algorithmic performance.

The numerical results with various budget levels are summarized in FIGURE 13. QGA(1), QGA(5) and QGA(10) indicate that the population size employed in QGA are 1, 5 and 10, respectively. For comparison purposes, we employ the population size of 10 in GA and use GA(10) to indicate this result in the same figure. It can be observed that GA(10) improves TSTT slowly in all cases. When budget levels φ are 100, 300, 500 and 900, GA hardly improves the TSTTs when reaching the pre-specified generation limit 10. On the other hand, obvious improvements in TSTTs can be seen across all cases in QGA(10). Though it starts with bad initial solutions in some cases, QGA(10) demonstrates dominant searching capability and quicker convergence when compared to GA(10). Note that φ is used to represent the total available budget in the numerical experiments.

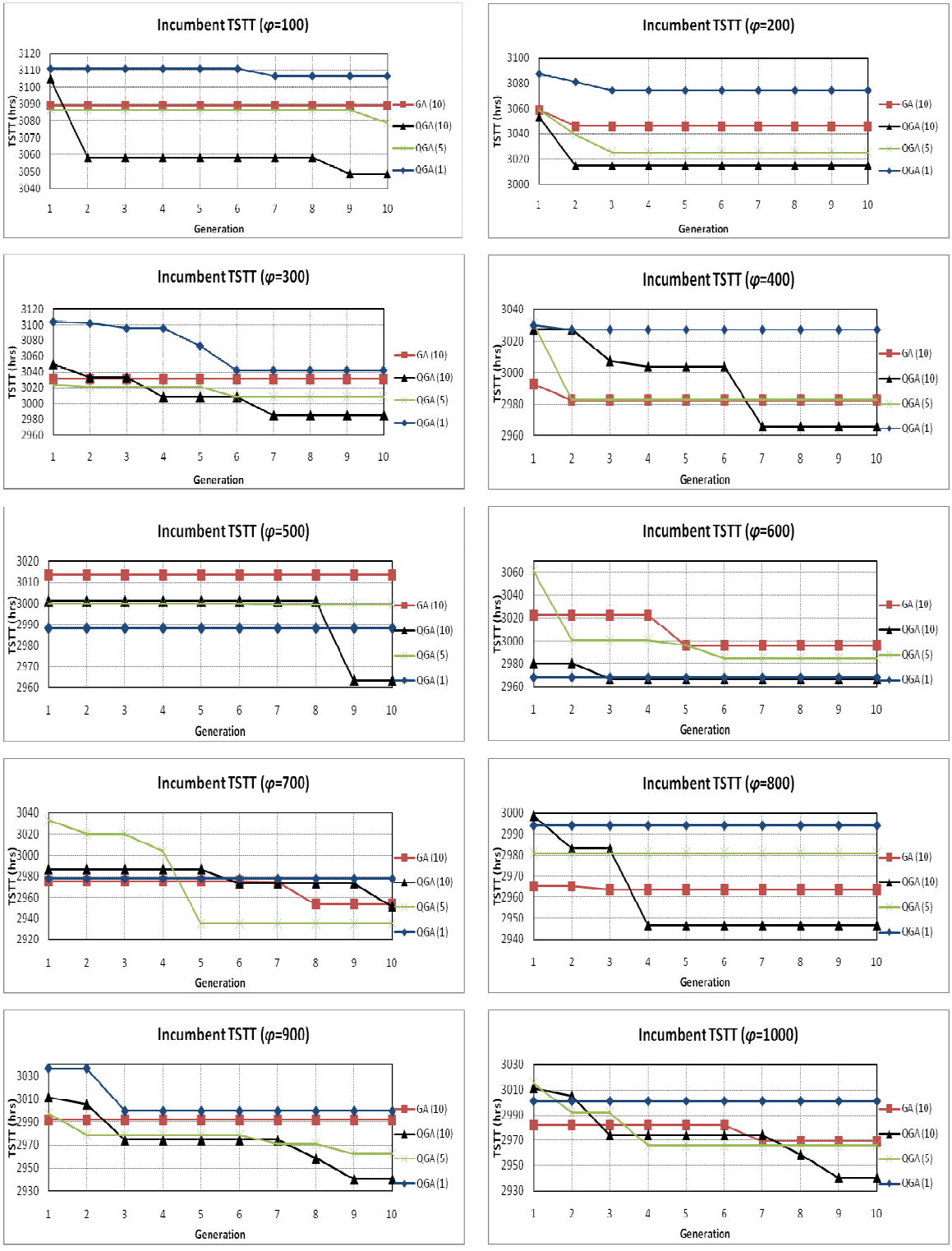

FIGURE 13: Numerical Results for Sioux Fall Network with Various Budget Levels

It should be noted that, except when $\varphi = 800$, QGA(5) outperforms GA(10) with half of the number of functional evaluations (50 versus 100). The computational benefit may not be obvious for small network tests, but could be critical for large-scale implementation. As mentioned before, from our experience, it takes over 48 hours to conduct one DTA simulation for the city of Austin in Texas. If the number of function evaluations can be reduced by half and solution quality is reasonably maintained, the computational savings could be tremendous. Interestingly, QGA(1) obtains a TSTT that is almost as good as that obtained by QGA(10) when $\varphi = 600$. If the convergence direction can be further calibrated for QGA(1), the solutions could potentially make this attractive solution strategy even more appealing.

It is obvious that GA needs one chromosome to represent one state (budget allocation in this dissertation). QGA, however, can represent multiple states with different probabilities in a single chromosome. This feature demonstrates QGA's superior ability in exploring the complex solution space. Therefore, it is expected that QGA would outperform GA under the same conditions (i.e. same population size and maximum number of generations), regardless of the GA parameters used.

8.3.2. Sensitivity Analysis of Congestion Level using Sioux Fall Network

Next, we test the performance of QGA with various congestion levels. To characterize the impact of the congestion levels on QGA, we conduct experiments with various OD demands which yield various congestion levels. In this experiment, we still limit the population size to 10, budget level to 1,000 units and the maximum number of generations to 10 for consistency. The results are summarized in FIGURE 14. As can be observed from the figure, when the congestion level is low (with $original \quad demand \times 0.5$), GA and QGA have identical performance. After verification, the link travel times in both the solutions obtained by GA and QGA are free-flow link

travel times. Therefore, both algorithms essentially find the optimal solution. Note that the total available budget is relatively high in this experiment. Thus, it is less demanding for a search procedure to find the optimal solution.

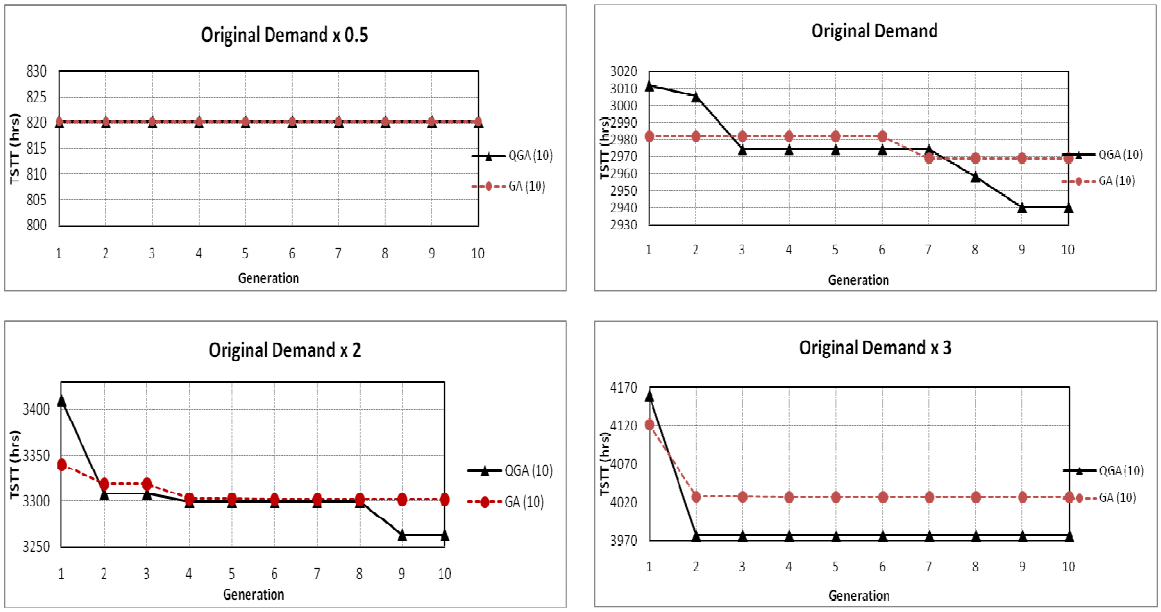

FIGURE 14: Numerical Results for Sioux Fall Network with Various Congestion Levels

When the OD demand increases to *original demand*, *original demand* $\times 2$ and *original demand* $\times 3$, QGA outperforms GA for these congested networks. In addition, the TSTT differences increase from 16.77 to 38.56 and 49.71 hours respectively. The relatively stringent budget level necessitates an accurate and efficient search procedure to identify the optimal capacity expansion policy. QGA in these experiments is able to find the budget allocation policies that benefit the system more than GA.

To calibrate the angles ($\Delta\theta_i$) of the Rotation Gate in QGA, replication runs with a large number of generations are usually required. For instance, Han and Kim (2002) employ the generation number of 1,000 to verify the angel selection process. In

addition, there are eight non-integer angles in the Rotation Gate that need to be calibrated. Potentially, one needs to examine all the combinations of these angles to identify the optimal angle combination. The number of angle combinations is easily intractable. Therefore, the computational efforts are prohibitive to adopt the above calibration process for dynamic transportation NDP. Thus, instead of fully calibrating the angles, we next conduct sensitivity analysis to demonstrate the impact of these angles on the convergence behaviors of QGA.

8.3.3. Sensitivity Analysis of Rotation Gate using Sioux Fall Network

In this section, we perturb the angles in the rotation gate and run the QGA(10) based on the modified angles. The results are summarized in TABLE 24.

TABLE 24: Incumbent TSTTs with Different Rotation Gate Angles (Hours)

Budget φ (Unit)	QGA(10) with $\Delta\theta_i$	QGA(10) with $2\times\Delta\theta_i$	QGA(10) with $\Delta\theta_i / 2$
100	3,048.45	3,057.03	3,048.35
200	3,014.92	3,009.58	3,002.03
300	2,985.43	2,980.19	2,985.38
400	2,965.93	2,997.29	2,962.97
500	2,963.53	2,964.39	2,966.38
600	2,967.24	2,967.34	2,967.04
700	2,951.64	2,955.15	2,948.34
800	2,946.26	2,970.46	2,936.71
900	2,940.48	2,957.04	2,978.25
1,000	2,954.45	2,940.15	2,952.67

By increasing the angles from $\Delta\theta_i$ to $2\times\Delta\theta_i$, the algorithm should tend to converge faster since the qubit chromosomes have a higher probability of moving toward the incumbent qubit chromosome. However, it does not necessarily mean that the solution quality will be always better by adopting $2\times\Delta\theta_i$, since the algorithm could potentially converge to a local optimum. By decreasing the angles from $\Delta\theta_i$ to

$\Delta\theta_i / 2$, on the other hand, the algorithm should have better chance of exploring a greater solution space and avoid getting stuck in local optimum. However, the solution quality could be compromised due to the limited number of iterations allowed. As can be seen, QGA (10) with $\Delta\theta_i / 2$ performs better than QGA (10) with $\Delta\theta_i$ in eight of the ten budget tests. In this comparison, further exploring the solution space is a better solution strategy. In contrast, QGA (10) with $2 \times \Delta\theta_i$ performs worse in eight of the ten budget levels. Local optimality may be the reason for the worse performance. Though it is difficult to draw a general conclusion based on the results above, the experiment does demonstrate that the angles employed in RG are not optimally calibrated to solve the dynamic network design problem and also reveals the possibility of further enhancing the search power of QGA.

8.3.4. Monticello Network

The second network (FIGURE 15) is the Monticello, Minnesota network from Xie (2008). The network contains 152 nodes and 306 links. After breaking the network according to CTM, the network consists of 9,372 cells that are candidates for capacity expansion. There are a total of 46 OD pairs, and the number of OD demands is 41,950 vehicles.

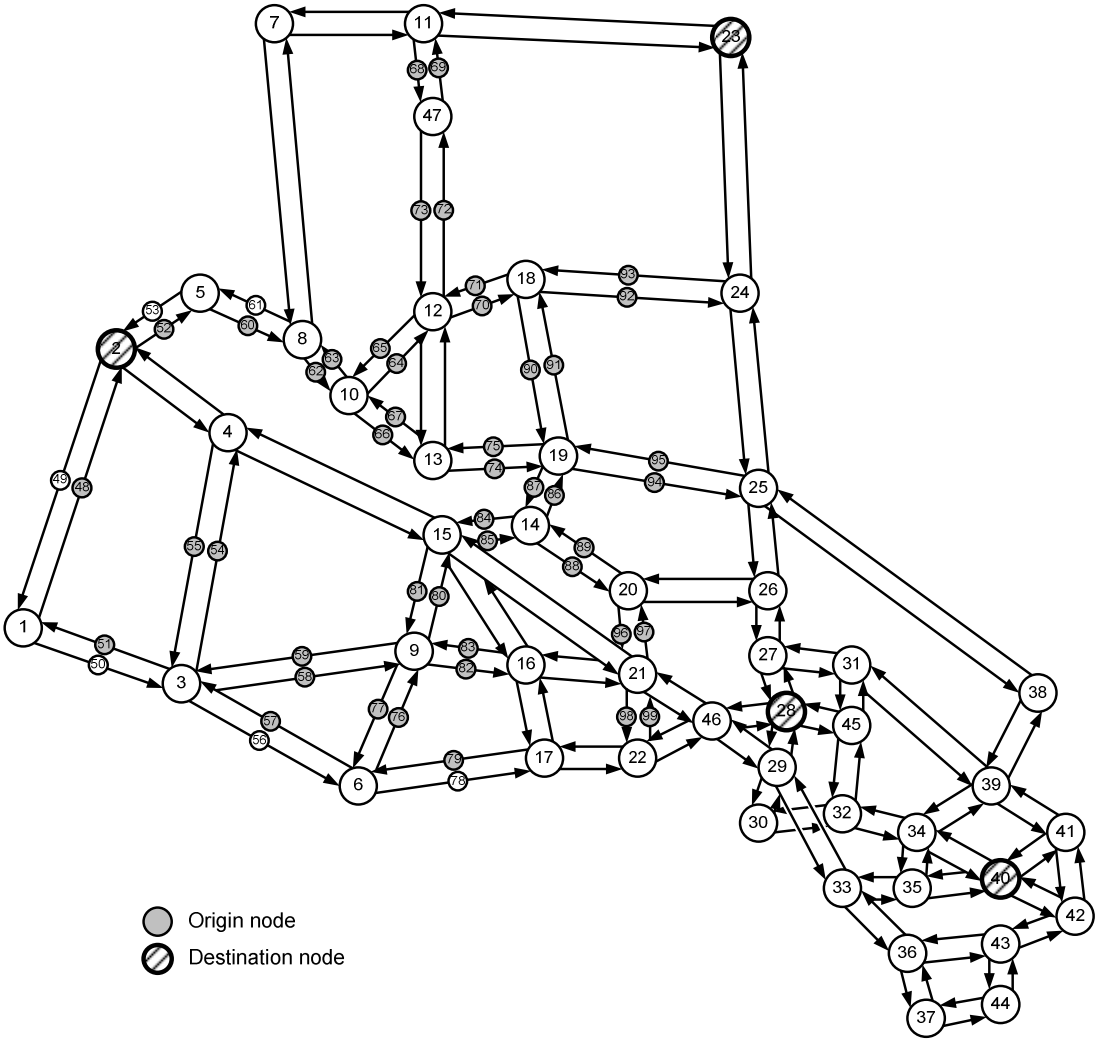

FIGURE 15: Monticello Network

The total available budget φ considered in this test is 1,000,000 units. Similarly, we employ VISTA as the evaluation function and the simulation duration is increased to 18,000 seconds to ensure that all vehicles leave the network after the planning horizon. Computational times required for network of this size for both GA (10) and QGA (10) are about 1,200 minutes, depending on the system performance when

conducting the experiment. The angles of RG are the angles from TABLE 23. The numerical results are summarized in FIGURE 16.

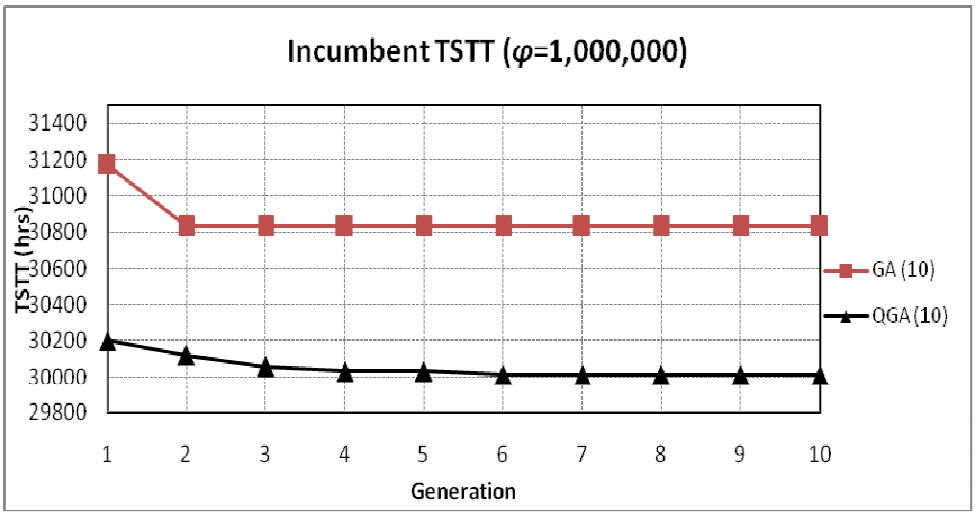

FIGURE 16: Numerical Results for Monticello Network

From FIGURE 16, we can see the QGA (10) dominates GA (10) in each generation for this large-scale network. Eventually, QGA (10) obtains the solution with TSTT 823.39 hours less than the solution obtained from GA (10). The main reason for QGA's superior performance is its ability to carry more information in a single qubit chromosome. Note that the computational time of the problem with this network size is about 20 hours. For the city of Austin, Texas, for instance, the number of links, nodes and OD demands are 19386, 9599 and 695013, respectively. To consider the dynamic network design problem of such a big city, it is not feasible to allow a huge number of populations and generations in genetic algorithms. This is when the QGA can really demonstrate its search power.

8.4 Summary

In this chapter, a floating-point encoded quantum-inspired genetic algorithm that uses insights gained through quantum computing is developed to solve the dynamic network design problem. A qubit chromosome that uses the qubit as the basic information unit is employed to represent the budget allocation. A rotation gate is utilized to update the qubit chromosome and a DTA module VISTA is employed to evaluate the budget allocation. From the preliminary results, the proposed QGA outperforms GA when population size and number of generations are limited to small numbers because of its ability to represent more than one state in each qubit chromosome. Even with a single qubit chromosome, QGA can potentially find good solutions. However, the direction of convergence is not fully calibrated in this chapter and needs to be further explored before QGA with single qubit chromosome reaches its full potential. The calibration of the angles in Rotation Gate can be a challenging task for the dynamic network design problem. As mentioned before, it generally takes repeated runs with a large number of generations to calibrate the angles used in the rotation gate. The process is extremely expensive and computationally prohibitive. This, therefore, indicates the first direction of future research. In addition, we do not employ the genetic operators such as crossover and mutation due to the probabilistic nature of the QGA. The possibility of improving the solution quality through the widely used genetic operators could be explored in future research.

In addition, we have shown that utilizing quantum logic can reduce the number of functional evaluations due to its ability to possess the superposition of states. For many transportation problems that are computationally difficult to solve, quantum logic opens up opportunities for developing various meta-heuristics (such as Tabu Search, Simulated Annealing, Ant Colony Optimization and Cross-entropy method) that are suitable for

different problems. The possibility of taking advantage of the superior representational power of quantum logic should be explored in future research.

Lastly, it should be emphasized that QGA, though it utilizes quantum mechanical principles, is not exactly a quantum algorithm. Numerous research efforts have been devoted to devising novel quantum algorithms, such as Shor's factorization algorithm. The design of a quantum algorithm that can be implemented and tested on a quantum computer is evolving and needs to be further investigated.

Chapter 9. Multiple-destination Bi-level Linear Programming Network Design Problem: A Descent Method

The objective of dynamic network design is to determine budget allocations in a network to maximize system performance when users route themselves in a selfish manner. In this chapter, we present the mathematical formulation of the continuous network design problem, with multiple-origin and multiple-destination O-D demands. The CTM, proposed by Daganzo (1994 and 1995), is employed to propagate traffic in the network. The formulation is an extension of the departure time-based linear programming DTA model (Li et al., 1999) and the Wardrop-type dynamic user equilibrium traffic assignment (Chang, 2004).

9.1 Mathematical Formulation

$$F\left[b, x(b), y(b)\right] = \underset{b,x,y}{Min} \sum_{\forall r \in C_R} \sum_{\forall s \in C_S} \sum_{\forall t \in T} \sum_{\forall i \in C \setminus \{C_S, C_R\}} x_{r,s,i}^{t} \tag{9.1}$$

subject to

$$\sum_{i \in C \setminus C_S} b_i \le TAB \tag{9.2}$$

$$b_i \ge 0 \qquad \forall i \in C \setminus C_S \tag{9.3}$$

$$\Psi(\Xi^*, b)^T (\Xi - \Xi^*) \ge 0 \qquad \forall \Xi \in K \tag{9.4}$$

where

$$
K=\left\{\begin{array}{ll}
x_{r,s,i}^{t}-x_{r,s,i}^{t-1}-\sum_{j\in P(i)}y_{r,s,ji}^{t}+\sum_{j\in S(i)}y_{r,s,ij}^{t-1}=d_{r,s,i}^{t} & \forall i\in C,\forall r\in C_R,\forall s\in C_S,\forall t\in T \quad (9.5.1)\\
\sum_{\forall j\in P(i)}y_{r,s,ij}^{t}-x_{r,s,i}^{t}\le 0 & \forall i\in C,\forall r\in C_R,\forall s\in C_S,\forall t\in T \quad (9.5.2)\\
\sum_{\forall r\in C_R}\sum_{\forall s\in C_S}\left(\sum_{\forall i\in P(i)}y_{r,s,ij}^{t}+x_{r,s,j}^{t}\right)\le \delta_j^t(N_j^t+\chi_j\cdot b_j) & \forall\forall j\in C\setminus(C_s,C_R),\forall t\in T \quad (9.5.3)\\
\sum_{\forall r\in C_R}\sum_{\forall s\in C_S}\left(\sum_{\forall j\in S(i)}y_{r,s,ij}^{t}\right)\le Q_i^t+\phi_i\cdot b_i & \forall i\in C\setminus C_s,t\in T \quad (9.5.4)\\
\sum_{\forall r\in C_R}\sum_{\forall s\in C_S}\left(\sum_{\forall i\in P(j)}y_{r,s,ij}^{t}\right)\le Q_j^t+\phi_j\cdot b_j & \forall j\in C\setminus C_s,t\in T \quad (9.5.5)\\
x_{r,s,i}^{0}=0 & \forall r\in C_R,\forall s\in C_S,\forall i\in C \quad (9.5.6)\\
y_{r,s,ij}^{0}=0 & \forall r\in C_R,\forall s\in C_S,\forall (i,j)\in E \quad (9.5.7)\\
x_{r,s,i}^{|T|}=0 & \forall r\in C_R,\forall s\in C_S,\forall i\in C \quad (9.5.8)\\
y_{r1,s,r2j}^{t}=0 & \forall r1,r2\in C_R,r1\ne r2,\forall r2j\in E,\forall s\in C_S,\forall t\in T \quad (9.5.9)\\
x_{r,s,i}^{t}\ge 0 & \forall r\in C_R,\forall s\in C_S,\forall i\in C,\forall t\in T \quad (9.5.10)\\
y_{r,s,ij}^{t}\ge 0 & \forall r\in C_R,\forall s\in C_S,\forall t\in T,\forall (i,j)\in E \quad (9.5.11)
\end{array}\right\}
$$

The objective function is the minimization of the TSTT, which can be calculated by summing up the vehicle occupancies for all cells other than the sink and source cells. Eq. (9.2) is the budget constraint, which restricts the allocated to be less than the Total Available Budget (TAB). Eq. (9.4) is the Variational Inequality (VI) of user-optimal dynamic traffic assignment (UODTA). Similarly, we can see that the UODTA flows, $\Xi^{'}$, always result in a lower total route cost than other feasible assignments ($\Psi(\Xi^{'},b)^{T}\Xi\ge\Psi(\Xi^{'},b)^{T}\Xi^{'}$). The solution space *K* of the VI is characterized by the CTM-related constraints (Eq. (9.5.1)-Eq. (9.5.11)). Eq. (9.5.1) is the cell mass flow conservation constraint. Eq. (9.5.2) limits the number of vehicles that leave a cell ($\sum_{\forall j\in P(i)}y_{r,s,ij}^{t}$) to be less than the number of vehicles currently in the cell ($x_{r,s,i}^{t}$). Eq. (9.5.3), Eq. (9.5.4), and Eq. (9.5.5) are the jam density, outgoing saturation flow rate, and incoming saturation flow rate of cells, respectively. Eqs. (9.5.6)-(9.5.7) state the initial cell conditions, while Eq. (9.5.8) states the final condition. Eq. (9.5.9) ensures that OD demands depart at the correct origins and arrive at the correct destinations. Non-negativity conditions are described in Eq. (9.5.10), Eq. (9.5.11), and Eq. (9.3).

Note that we view the formulation solely in terms of the upper-level variable $b_i \quad \forall i \in C \setminus C_S$. The lower-level variables (*x*,*y*) are functions of the upper-level variable. This idea has been adopted in many descent-based approaches (Vicente and Calamai, 1994). In a recent effort, this approach is applied to an OD matrix estimation problem (Lundgren and Peterson, 2008). The major issue with a descent method is that the upper-level objective function may not be differentiable at every feasible point. In other words, information about the gradient may not be available at all feasible points (Colson et al., 2007)). Therefore, to employ this method effectively, we devise a gradient approximation scheme to obtain network-wide gradient information in an efficient manner.

9.2 A Descent Method

The descent method devised in this dissertation treats the bi-level program solely in terms of the upper-level variables (b_i), and considers lower-level variables ($x_{r,s,i}^t$ and $y_{r,s,ij}^t$) to be functions of the upper variables. The method attempts to find a feasible direction along which the upper-level objective decreases, given a feasible lower-level solution. To be specific, we present the following pseudo code, followed by detailed explanations.

Step 0

$b_i = 0 \qquad \forall i \in C \setminus C_S$

Initial step length λ

Iter = 1

Step 1

Update cells in the CTM

$$N_i^t \leftarrow N_i^{t,original} + \chi_i \cdot b_i \qquad \forall i \in C \setminus C_S$$

$$Q_i^t \leftarrow Q_i^{t,original} + \phi_i \cdot b_i . \qquad \forall i \in C \setminus C_S$$

Step 2

Conduct user-optimal dynamic traffic assignment to evaluate *TSTT*. If *TSTT* fails to improve in three consecutive iterations, $\lambda \leftarrow \frac{\lambda}{2}$

Step 3

Apply the chain rule to $F[b, x(b), y(b)]$ to find the descent direction: *D*:

$$D = \nabla_b F[b, x(b), y(b)] + \nabla_x F[b, x(b), y(b)] \cdot \nabla_b x + \nabla_y F[b, x(b), y(b)] \cdot \nabla_b y$$

$$= (\frac{\partial F}{\partial b}) + (\frac{\partial F}{\partial x}) \cdot (\frac{\partial x}{\partial b}) + (\frac{\partial F}{\partial y}) \cdot (\frac{\partial y}{\partial b})$$

$$= (\frac{\partial F}{\partial b_1}, \frac{\partial F}{\partial b_2}, ..., \frac{\partial F}{\partial b_n}) +$$

$$(\frac{\partial F}{\partial x^1_{1,1,1}}, ..., \frac{\partial F}{\partial x^t_{r,s,n}}) \cdot \begin{bmatrix} \frac{\partial x^1_{1,1,1}}{\partial b_1} & \cdots & \frac{\partial x^1_{1,1,1}}{\partial b_n} \\ \cdots & \cdots & \cdots \\ \frac{\partial x^t_{r,s,n}}{\partial b_1} & \cdots & \frac{\partial x^t_{r,s,n}}{\partial b_n} \end{bmatrix} + (\frac{\partial F}{\partial y^1_{1,1,12}}, ..., \frac{\partial F}{\partial y^t_{r,s,ij}}) \cdot \begin{bmatrix} \frac{\partial y^1_{1,1,12}}{\partial b_1} & \cdots & \frac{\partial y^1_{1,1,12}}{\partial b_n} \\ \cdots & \cdots & \cdots \\ \frac{\partial y^t_{r,s,ij}}{\partial b_1} & \cdots & \frac{\partial y^t_{r,s,ij}}{\partial b_n} \end{bmatrix}$$

Step 4

Update incumbent solution: $b_i^{iter+1} = b_i^{iter} + \lambda_i \cdot D_i \qquad \forall i \in C \setminus C_S$

If $\sum_{i \in C \setminus C_S} b_i > TAB$, $b_i \leftarrow \frac{b_i}{\sum_{i \in C \setminus C_S} b_i} \times TAB$

Step 5

If

All components in descent direction *D* are 0 or *Iter* > MAX_ITERATION, stop.

End if

Else

$Iteration \leftarrow Iteration + 1$

go to Step 1.

Step 0 is the initialization step, which initializes the budget allocation policies (b_i), step length (λ), and iteration counters (*iter*). Step 1 updates the jam densities and saturation flow rates of cells, based on the current budget allocation. The approach then conducts a UODTA to obtain the *TSTT*, the performance measure of the current budget allocation. If the *TSTT* fails to improve in three consecutive iterations, the step length is halved. The UODTA also gives feasible value of the lower-level variables, which are then used in subsequent steps to approximate the gradient/descent information. The chain rule is applied in Step 3 to obtain the descent direction (*D*). The approximation

scheme for the components in *D* will be presented in the following section. After finding the direction, budget allocation policies are updated accordingly, based on the direction and step size. Note that budget allocation at this step can exceed the total available budget. If this occurs, we proportionally reduce the budget allocation. Step 5 checks the stopping criteria. If all elements in *D* are zero, the budget allocation will remain the same in further iterations, and thus, further iterations cannot improve the *TSTT* of the system. Therefore, this serves as a stopping criterion, and the algorithm terminates when it is met. The other stopping criterion is the iteration limit. The approach will be terminated upon reaching the predefined iteration limit. The next step is to obtain the descent direction. As there is not an existing approach that can calculate the descent direction detailed in Step 3, we propose to break the descent direction into basic elements, and approximate each element separately.

9.3 Descent Direction Approximation

There are essentially five basic elements in the descent direction. In this section, we present the approximation scheme for each element. The elements obtained are employed as the descent direction, within the framework of the algorithm.

9.3.1. Approximation of $\frac{\partial F}{\partial b_i}$

The interpretation of this value is the change in the *TSTT* due to the change of the budget allocated to cell *i*. In this research, we approximate this value from the single level dynamic System Optimal Network Design Problem (SONDP) (Eqs. (9.1)-(9.3) and Eqs.(9.5.1)-Eq.(9.5.11)). Note that Eqs. (9.5.3)-(9.5.5) are the constraints, with budget allocation variable b_i on the right-hand-side. We then choose to employ the sum of the dual variables of Eqs.(9.5.3)-(9.5.5) as the approximated value of $\frac{\partial F}{\partial b_i}$. However,

instead of solving SONDP and obtain the corresponding decision variables, we employ the $x_{r,s,i}^t$ and $y_{r,s,ij}^t$ from UODTA, to ensure that the cell occupancy and flow rate come from user-optimal behaviors.

The dual variables of Eqs.(9.5.3)-(9.5.5) can be obtained from the complementary slackness properties when we consider the pure linear program of SONDP. For instance, if $\sum_{\forall r\in C_R}\sum_{\forall s\in C_S}\left(\sum_{\forall i\in P(i)} y_{r,s,ij}^t + x_{r,s,j}^t\right) = \delta_i^t(N_j^t + \chi_i \cdot b_i)$ in Eq. (9.5.3), then the corresponding dual variable is zero. Similarly, we can construct the dual program of the SONDP and apply the complementary slackness properties to it, provided that this constraint is not binding in the primal formulation. The dual program of the SONDP is referred to in Li et al. (1999) and the detailed dual approximation approach using the complimentary slackness condition can be found in earlier chapters. Similarly, for Eq.(9.5.4) and Eq.(9.5.5), we can apply the same technique and efficiently obtain the corresponding dual variables.

Note that the abovementioned approach is simply one way of approximating the value of $\frac{\partial F}{\partial b_i}$. In a method similar to the re-simulation dual approximation technique proposed earlier, one can run two UODTAs to approximate $\frac{\partial F}{\partial b_i}$ with greater accuracy. In the first UODTA, one can obtain *TSTT* by evaluating UODTA. Then, the budget allocated to cell *i* should be perturbed by one unit, and the capacity of that cell should be changed accordingly. Solving UODTA with the updated cell capacity, one can obtain the new *TSTT'*. Then $\frac{\partial F}{\partial b_i} = TSTT' - TSTT$. However, the exact approach requires prohibitive computational efforts and limits its practical use, especially when applied to a

large-scale problem. Therefore, to improve the computational efficiency, we adopt the approximation process, which facilitates the complimentary slackness conditions.

9.3.2. Approximation of $\frac{\partial F}{\partial x_{r,s,i}^t}$

This partial derivative is interpreted as the change in TSTT due to the change in occupancy of cell *i* during time interval *t*, when the change results from the additional user departing from cell *r* and arriving at cell *s*. In this dissertation, we propose to approximate this value with the following approach. First, the additional demand destined to cell *s* is added to cell *i* at time interval *t*. Then, we identify the TDSP for that demand. The travel time of this TDSP is taken as this value.

One can adopt the re-simulation approach mentioned in earlier sections in a similar manner. Again, using the re-simulation approach can limit the applicability of the proposed heuristic. Therefore, we choose to apply the approximation scheme, so the heuristic can potentially solve larger problems.

9.3.3. Approximation of $\frac{\partial x_{r,s,i}^t}{\partial b_i}$

The value of this element can be interpreted as the change of cell occupancy due to the budget allocation change. If cell *i* is not congested at time interval *t*, investing budget to this cell should not change the cell occupancy. Thus, we can approximate the value of $\frac{\partial x_{r,s,i}^t}{\partial b_i}$ using the following two facts.

Fact1. $\frac{\partial x_{r,s,i}^t}{\partial b_i} = 0$ for an uncongested cell.

Fact2. $\frac{\partial x_{r,s,i}^{t}}{\partial b_i} = 1$ for a congested cell *i*, provided that there exists at least one vehicle in the upstream cells of cell *i* at time interval *t-1*.

9.3.4. Approximation of $\frac{\partial F}{\partial y_{r,s,ij}^{t}}$

This value corresponds to the change in the total system travel time with respect to the change of the flow from cell *i* to cell *j*, at time interval *t*. Therefore, the approximation of this value is similar to $\frac{\partial x_{r,s,i}^{t}}{\partial b_i}$. However, instead of adding additional demand to cell *i* at time interval *t* for this value, we add the demand at cell *j* at time interval *t*, since we are approximating this value for the inflow of cell *j*. The approximated value is the travel time of the TDSP for a vehicle from cell *j* to its destination *s*, departing at time interval *t*.

9.3.5. Approximation of $\frac{\partial y_{r,s,ij}^{t}}{\partial b_j}$

This value can be interpreted as the change of flow between cell *i* and cell *j* due to the budget allocation change. This value can be obtained utilizing the following facts.

Fact 3. $\frac{\partial y_{r,s,ij}^{t}}{\partial b_j} = 0$ if cell *j* is not congested at time interval *t*,

Fact 4. $\frac{\partial y_{r,s,ij}^{t}}{\partial b_j} = 1$ for a congested cell *j*, provided that there exists at least one vehicle in the upstream cells of cell *j* at time interval *t-1*.

9.4 Numerical Experiments

To obtain better insights, we illustrate the descent method using two numerical experiments, on two networks of differing sizes.

9.4.1. Effectiveness

The first network is a 6-cell CTM network, depicted in FIGURE 2. The data regarding this network are presented in TABLE 1 and TABLE 2. The descent method and required functions are implemented in the standard JAVA language. The numerical experiments are conducted on a Linux machine with an Intel 3.00GHz CPU and 32 GB of memory.

For comparison, we also present the optimal solutions from the modified K^{th}-Best algorithm (Karoonsoontawong and Waller, 2006). The performance measure is based on the total system travel time (TSTT), evaluated from the objective function ($\sum_{\forall r \in C_R} \sum_{\forall s \in C_S} \sum_{\forall t \in T} \sum_{\forall i \in C \setminus \{C_S, C_R\}} x_{r,s,i}^t$). Performance is better if the TSTT is lower, given the same budget level. The computational results are presented in TABLE 25. It can be observed that when the budget level is high, both methods obtain an identical solution in terms of TSTT. However, when a stringent budget is considered, the K^{th}-Best Algorithm marginally outperforms the proposed descent method. The average optimality gap in this experiment is 2.7%, which demonstrates the effectiveness of the proposed descent method.

TABLE 25: 6-cell CTM Network Cell Expansion Policies with Different Budgets

TAB (unit)	Descent Method						Optimal K^{th}-Best Algorithm					
	τ	b[1]	b[2]	b[3]	b[4]	b[5]	τ	b[1]	b[2]	b[3]	b[4]	b[5]
50	14.00	2.87	0	8.25	1.18	37.71	14.00	2.00	0	0	0	5.00
40	14.00	2.83	0	6.50	0.93	29.73	14.00	2.00	0	0	0	5.00
30	14.00	2.78	0	4.76	0.68	21.77	14.00	2.00	0	0	0	5.00
20	14.00	2.68	0	3.03	0.43	13.85	14.00	2.00	0	0	0	5.00
10	14.00	2.42	0	1.33	0.19	6.06	14.00	2.00	0	0	0	5.00
9	14.00	2.37	0	1.16	0.17	5.30	14.00	2.00	0	0	0	5.00
8	14.40	0.16	0	1.12	0.16	5.12	14.00	2.00	0	0	0	5.00
7	15.10	1.56	0	0.95	0.14	4.36	14.00	2.00	0	0	0	5.00
6	15.90	1.5	0	0.79	0.11	3.60	15.00	2.00	0	0	0	4.00
5	16.73	1.43	0	0.63	0.09	2.86	16.00	2.00	0	0	0	3.00
4	17.18	1.23	0	0.15	0.02	2.60	17.00	2.00	0	0	0	2.00
3	18.01	0.94	0	0	0	2.06	18.00	1.50	0	0	0	1.50
2	19.75	0.43	0	0	0	1.57	19.00	1.00	0	0	0	1.00
1	23.42	0.09	0	0	0	0.91	21.00	0.333	0	0	0	0.667

However, it should be emphasized that the K^{th}-Best Algorithm is not suitable for large-scale implementation, though it does provide useful insights for analyzing the problem. This intrinsic limitation constrains the usefulness of this method for large-scale application. Scalability and flexibility are the advantages of the descent method. For instance, one can employ cells with larger sizes to improve the algorithmic performance. In addition, the approximation scheme of the descent method can be applied at a link level or even a regional level, instead of the current cell level. This could potentially improve the performance and increase the applicability of the method.

9.4.2. Efficiency

From the experiment above, it can be seen that the descent method can potentially solve the dynamic NDP, and obtain the solution with an optimality gap of 2.7%. In the

following experiment, we demonstrate the efficiency of the descent method with a 68-cell CTM network. The CTM network and details are illustrated in FIGURE 3,

TABLE 4, and TABLE 5. In this experiment, we rely on the previous work of Karoonsoontawong and Waller (2006), who demonstrated through extensive numerical testing that, among various meta-heuristic approaches, the specific GA implementation we used performs very well for this specific problem. The results are depicted in FIGURE 17.

We compare our result for 68-cell CTM networks with GA solutions. We limit the number of iterations of the descent method to be 10 and the budget level to be 50 in this experiment. It can be seen that the proposed descent method finds the better solution in the first few iterations. However, further improvement is simply marginal. After the first few iterations, the method cannot further improve the *TSTT*. On the other hand, the GA starts with high *TSTT* and gradually converges to the solution found by the descent method. Though the *TSTT* keeps improving in the GA, the convergence is much slower than the proposed descent method.

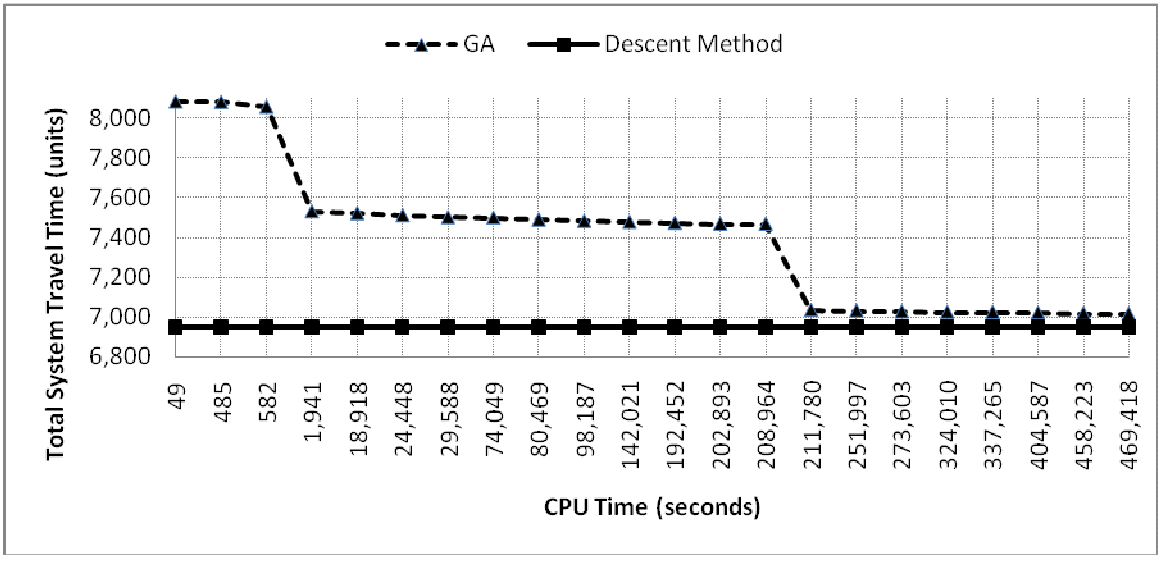

FIGURE 17: 68-cell CTM Numerical Experiment

The CPU time required by the descent method is 2,428 seconds, and the *TSTT* obtained is 6,947 units. To obtain a comparable TSTT (*TSTT*=7,016), the GA requires 469,418 seconds, which is roughly 65 times the CPU time required by the descent method. This observation empirically proves the efficiency of the descent method.

9.5 Summary

A descent method that explores the mathematical programming structure of multiple-origin and multiple-destination BLPNDP is presented in this chapter. The proposed method identifies the dual variables (partial derivatives) of the associated constraints and constructs the descent direction based on those variables. Numerical experiments have demonstrated both the accuracy and efficiency of the proposed method. However, BLPNDP with larger network sizes should be tested and examined to show the applicability of the proposed method to real-world problems.

Chapter 10. Conclusions and Future Extensions

This dissertation proposes bi-level formulations of various transportation problems, including dynamic network design, dynamic congestion pricing, and off-line dynamic traffic assignment capacity calibration. To efficiently solve the problems, scalable dual variable approximation techniques are presented, and experiments are performed on networks of various sizes. For specific problems, we propose different solution strategies based on the dual variable approximation techniques, which can serve as a system-wise gradient.

Dantzig-Wolfe Decomposition based Heuristic Scheme for BLPNDP

BLPNDP is NP-hard by nature. To solve this problem, we employ the dual variable approximation techniques, and embed them into the Dantzig-Wolfe decomposition scheme. The decomposed problems include a restricted master program and a series of pricing programs. The restricted master program, which is essentially a budget allocation problem, determines the budget allocation policies and modifies network capacities accordingly. The pricing program, UODTA, finds the dynamic traffic assignment that incorporates selfish routing of road users and passes the dual variables back to the restricted master programs, so that the budget allocation policies can be revised according to user responses. The proposed solution scheme is tested with networks of various sizes. The heuristic solutions are compared with both the exact solution for a small network and the meta-heuristic solutions for a large network. From the numerical experiments, the proposed heuristic efficiently solves the problem with promising solution quality.

Dantzig-Wolfe Decomposition based Heuristic Scheme for Off-line Dynamic Traffic Assignment Capacity Calibration

In this dissertation, we present the mathematical formulation of the off-line dynamic traffic assignment capacity calibration problem, which calibrates network capacities such that the counts predicted by the DTA module match the counts observed in the field. By reformulating the presented mathematical formulation, we propose a Dantzig-Wolfe decomposition-based heuristic scheme that decomposes the problem into a restricted master program and pricing programs. The restricted master program calibrates the capacities based on the dual information passed from the pricing program. The pricing program solves the dynamic traffic assignment with the calibrated capacities passed from the restricted master program. The iterative process repeats until a stopping criterion is met. From the numerical experiments conducted, we show that the proposed heuristic can calibrate the network capacity and match the counts within a 1% optimality gap.

Method of Successive Average Algorithm for Dynamic Congestion Pricing

By exploiting the meaning of approximated dual variables, we embed the dual variable approximation scheme into dynamic congestion pricing framework. Essentially, the dual variables of CTM-related constraints can be interpreted as the externality imposed by road users. Thus, the dual variables can be used to determine the time-varying tolls levied in the network. We propose a heuristic that is able to capture user behavior with imposed tolls. With appropriate algorithmic design and approximated toll values, we demonstrate that the solution framework can drive user-optimal behavior toward system-optimal behavior, so that the system performance is optimized.

Meta-heuristics

Two meta-heuristic algorithms-a quantum-inspired genetic algorithm and a dual approximation genetic algorithm-are presented in this dissertation to tackle the NP-complete bi-level problems considered herein. The quantum-inspired genetic algorithm seeks to incorporate the concept of quantum computing into the framework of conventional genetic algorithm. By simulating qubits and employing rotation gate in the genetic algorithm framework, a complex solution space can be effectively captured and efficiently explored. In numerical experiments on networks of various sizes, the quantum-inspired genetic algorithm presented in this dissertation performs better than the conventional genetic algorithm, in both efficiency and solution quality.

Descent Method

To facilitate the developed dual variable approximation techniques, and extend those techniques to multiple-origin, multiple-destination bi-level network design problem, we present a descent method that considers the lower-level variables to be functions of the upper-level variables, and apply the chain rule to identify the descent direction. By incorporating dual variable approximation techniques, we are able to construct the descent direction using the dual variables or the observed facts. Using numerical experiments, we have demonstrated the efficacy and efficiency of the proposed descent method.

Future Extensions

We will discuss the potential future extensions in this section. The first future extension is the multiple-destination problem of the off-line DTA capacity calibration and

dynamic congestion pricing problems. As the proposed descent method in chapter 9 is general in nature, we should be able to tackle these two problems in a similar manner.

The second potential extension is the optimal approach to DTA calibration, which considers demand and supply calibration. This dissertation solely examines the capacity calibration of DTA to facilitate the eventual development of multi-variable system-wide calibration methodologies, which refer to methods that consider demand, supply, erroneous traffic counts, and non-standard behavioral assumptions; such methods clearly required long-term effort. In addition, only the off-line calibration of DTA is considered in this dissertation; on-line DTA calibration should also be considered, as it could be helpful for real-time traffic management.

The next potential extension is the integrated dynamic traffic management framework. It is evident that significant advancements have occurred in signal design optimization, capacity expansion planning, and dynamic congestion pricing. However, the progress in these research streams has been achieved relatively independently. In addition, due the complexity of the problem, it is not uncommon to decompose the seemingly integrated problem into subproblems, as is always done in the four-step transportation planning process. However, problem decomposition is apparently not ideal, since it can lead to suboptimal deployment plans that can significantly reduce the system performance. To address these challenges, it is imperative to develop a conceptually unified and scalable framework that draws from the advantages revealed in advanced research in these fields. Development of such a framework would require dedicated, long-term research efforts devoted to studying each aspect of the problem.

The last potential extension is quantum-inspired meta-heuristics. In an earlier section, a QGA was devised to solve the bi-level dynamic network design. Though it performs better than a baseline genetic algorithm in the preliminary experiments, a QGA

can potentially get stuck in local optima due to its fast convergence. Therefore, one of the potential extensions is to combine QGAs with the concept originated by the Tabu search algorithm to develop a hybrid meta-heuristic algorithm. As in the Tabu search, one may keep a tabu list in the QGA solution framework, so that no solution can be revisited within a predefined number of iterations. This modification could prevent stoppage at local optima, and allow better solutions in the neighborhood to be explored. Similarly, incorporating quantum logic into various meta-heuristics (e.g., Simulated Annealing, Ant Colony Optimization, Random Search, and Cross-entropy method) is an exciting area for exploration.

References

1. Abdulaal, M., and L. J. LeBlanc. (1979). Continuous Equilibrium Network Design Models. *Transportation Research B*, Vol. 13, pp.19-32.
2. Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. (1993). *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, New Jersey.
3. Arnott, R., A. de Palma and R. Lindsey (1990). Economics of a Bottleneck. *Journal of Urban Economics* 27 pp. 11-30.
4. Arnott, R. and Small, K. (1994). The Economics of Traffic Congestion. *American Scientists*, 20(2), pp. 123–127.
5. Arnott, R. and Kraus, M. (1998). When are Anonymous Congestion Charges Consistent with Marginal Cost Pricing. *Journal of Public Economics* 67, pp. 45–64.
6. Baker, J. E. (1987) Reducing bias and inefficiency in the selection algorithm. In Genetic Algorithms and Their Applications: *Proceedings of the Second International Conference on Genetic Algorithms*, Massachusetts Institute of Technology, Hillsdale, N.J., pp. 28-31.
7. Balakrishna, Ben-Akiva, R.M. and Koutsopoulos, H.N. (2007) Offline Calibration of Dynamic Traffic Assignment: Simultaneous Demand-and-Supply Estimation. *Transportation Research Record*, No. 2003, pp. 50-58.
8. Benioff, P. (1980) The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines. *Journal of Statistical Physics*, Vol.22, No. 5, 563-591.
9. Bard J.F. (1983) An efficient point algorithm for a linear two-stage optimization problem. *Operations Research*, 31(4):670-684.
10. Bard, J. F. (1998). *Practical Bilevel Optimization Algorithms and Applications*, Kluwer Academic Publishers.
11. Boyce, D. E. (1984). Urban Transportation Network Equilibrium and Design Models: Recent Achievements and Future Prospectives. *Environment and Planning A*, Vol. 16, pp. 1445-1474.
12. Braid, R.M. (1996). Peak-load Pricing of a Transport Facility with an Unpriced Substitute. *Journal of Urban Economics* 40, pp. 179–197.
13. Bureau of Public Roads (1964). *Traffic Assignment Manual*. U.S. Dept. of Commerce, Urban Planning Division, Washington D.C.

14. Cascetta, E. and Cantarella, G.E. (1991) A Day-to-day and Within-day Dynamic Stochastic Assignment Model, *Transportation Research Part A*, Vol. 25, Issue 5, pp.277-291.

15. Cascetta, E. and Postorino, M.N. (2001) Fixed Point Approaches to the Estimation of O/D Matrices Using Traffic Counts on Congested Networks, *Transportation Science*, Vol. 35, pp. 134-147.

16. Cantarella, G.E., Pavone, G. and Vitetta, A. (2006) Heuristics for urban road network design: Lane layout and signal settings. European Journal of Operational Research Vol. 175, Issue 3, pp. 1682–1695.

17. Carey, M. and Srinivasan, A. (1993). Externalities, Average and Marginal Costs, and Tolls on Congested Networks with Timevarying Flows. *Operations Research* 41 (1), pp. 217–231.

18. Chang, E. J. (2004). Time-Varying Intermodal Person Trip Assignment. *Ph.D. Dissertation*, Northwestern University.

19. Chen, L. and May, A. D. (1987) Traffic detector errors and diagnostics. *Transportation Research Record*, No. 1132, pp. 82–93.

20. Chen, M., and A. S. Alfa. (1991). A Network Design Algorithm Using a Stochastic Incremental Traffic Assignment Approach. *Transportation Science*, Vol. 25, No. 3, pp. 215-224.

21. Chen, C., Kwon, J., Rice, J. Skabardonis, A. and Varaiya, P. (2003). Detecting errors and imputing missing data for single-loop surveillance systems. *Transportation Research Record*, No. 1855, 160–167.

22. Chu, X. (1995). Endogenous Trip Scheduling: the Henderson Approach Reformulated and Compared with the Vickrey Approach. *Journal of Urban Economics* 37 pp. 324-343.

23. Colson, B., Marcotte, P. and Savard, G. (2007). An Overview of Bilevel Optimization, *Annals of Operations Research*, Vol. 153, pp. 235-256.

24. Daganzo, C.F. (1994). The Cell Transmission Model: A Dynamic Representation of Highway Traffic Consistent with the Hydrodynamic Theory. *Transportation research Part B*, Vol. 28B, No.4, pp. 269-287.

25. Daganzo, C.F. (1995). The Cell Transmission Model, Part II: Network Traffic. *Transportation Research Part B*, Vol. 29B, No. 2, pp. 79-93.

26. Dantzig, G. B. (1963). *Linear Programming and Extensions*, Princeton University Press.

27. Dantzig, G. B., R. P. Harvey, Z. F. Landsowne, D. W. Robinson, and S. F. Maier. (1979) Formulating and Solving the Network Design Problem by Decomposition. *Transportation Research B*, Vol. 13, pp. 5-17.

28. Davis, G. A. (1994). Exact Local Solution of the Continuous Network Design Problem via Stochastic User Equilibrium Assignment. *Transportation Research B*, Vol. 28, pp. 61-75.

29. De Palma, A. and Lindsey, R. (2000). Private Toll Roads: Competition under Various Ownership Regimes. *Annals of Regional Science* 34 (1), pp. 13–35.

30. De Palma, A., Kilani, M., and Lindsey, R. (2005). Congestion Pricing on a Road Network: A Study using the Dynamic Equilibrium Simulator METROPOLIS. *Transportation Research Part A*, 39, pp. 588-611.

31. DiVincenzo, D. P. (1995) Quantum Computation. *Science*, Vol. 270, No. 5234, 255-261.

32. Ferrari, P. (2002). Road Network Toll Pricing and Social Welfare. *Transportation Research, Part B,* 36 (5), pp. 471–483.

33. Friesz, T. L. (1985). Transportation Network Equilibrium, Design and Aggregation: Key Developments and Research Opportunities. *Transportation Research A*, Vol. 19, pp. 413-427.

34. Friesz, T. L., H. Cho, N. Mehta, R. Tobin, and G. Anandalingam. (1992). A Simulated Annealing Approach to the Network Design Problem with Variational Inequality Constraints. *Transportation Science*, Vol. 26, No. 1, pp. 18-26.

35. Giraldi, G.A., R. Portugal and R. N. Thess. (2004) Genetic Algorithm and Quantum Computation. *arXiv:cs/0403003*.

36. Golani and Waller (2004). Combinatorial Approach for Multiple-Destination User Optimal Dynamic Traffic Assignment. *Transportation Research Record*, No. 1882, pp. 70–78.

37. Grefenstette, J.J. (1990) *A User's Guide to GENESIS Version 5.0*.

    http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/genetic/ga/systems/genesis/ (accessed 04/08)

38. Grover, L.K. (1996) A Fast Quantum Mechanical Algorithm for Database Search. *Annual ACM Symposium on Theory of Computing: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 212-219.

39. Han, K.-H., Park, K.-H., Lee, C.-H. and Kim, J.H. (2001) Parallel Quantum-Inspired Genetic Algorithm for Combinatorial Optimization Problem. *Proceedings of Evolutionary Computation*, 1422-1429.

40. Han, K.-H., and J.-H. Kim. (2002) Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization. *IEEE Transactions on Evolutionary Computation* Vol. 6, No. 6, 580-593.

41. Han, K-.H. and Kim, J.-H, (2003) On Setting the Parameters of Quantum-Inspired Evolutionary Algorithm for Practical Application. Proceedings of Evolutionary Computation, 178-184.

42. He, R. R., Ran, B. (2000). Calibration and Validation of a Dynamic Traffic Assignment Model. *Transportation Research Record*, No. 1733, pp. 56–62.

43. Hearn, D.W. and Yildirim, M.B. (2001). A Toll Pricing Framework for Traffic Assignment Problems with Elastic Demands. *Current Trends in Transportation and Network Analysis*: Miscellanea in Honor of Michael Florian, M. Gendreau, P. Marcotte (eds.), Kluwer Academic Publishers, Dordrecht, The Netherlands.

44. Henderson, J.V. (1974). Road Congestion: a Reconsideration of Pricing Theory. *Journal of Urban Economics* 1 pp. 346-365.

45. Henderson, J.V. (1981). The Economics of Staggered Work Hours. *Journal of Urban Economics* 9 pp. 349-364.

46. Hoang, H. H. (1982). Topological Optimization of Networks: A Non-Linear Mixed Integer Model Employing Generalized Benders Decomposition. *IEEE Transactions on Automated Control*, Vol. 27, pp. 164-169.

47. Holland, J. H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.

48. Hooke, R., and Jeeves, T. A. (1961) Direct search solution of numerical and statistical problems. *Journal of the ACM*, Vol. 8, pp. 212-229.

49. IBM Almaden Research Center (2000), http://www.almaden.ibm.com/. (Accessed October/2008)

50. Ishak, S., Alecsandru, S.C. and Seedah, D. (2006) Improvement and Evaluation of Cell-Transmission Model for Operational Analysis of Traffic Networks: Freeway Case Study. *Transportation Research Record*, No. 1965, pp. 171-182.

51. Janson, B. N. (1995). Network Design Effects of Dynamic Traffic Assignment. *Journal of Transportation Engineering*, Vol. 121, No. 1, pp. 1-13.

52. Jeon, K., Ukkusuri, S. and Waller, S. T. (2005). Heuristic Approach for Discrete Network Design Problem Accounting for Dynamic Traffic Assignment Conditions: Formulations, Solution Methodologies, Implementations and Computational Experiences. Presented at 84th Annual Meeting of the Transportation Research Board, Washington, D.C.

53. Jha, M., Gopalan, G., Garms, A., Mahanti, B. P., Toledo, T., and Ben-Akiva, M. E. (2004) Development and Calibration of a Large-Scale Microscopic Traffic Simulation Model, *Transportation Research Record*, No. 1876, pp. 121–131.

54. Johnson, D.S., Lenstra, J. K. and Rinnooy Kan, A. H. G. (1978) The Complexity of the Network Design Problem, *Networks*, Volume 8, Issue 4, 279-285.

55. Joksimovic D., Bliemer, M.C.J and Bovy, P.H.L. (2005). Optimal Toll Design Problem In Dynamic Traffic Networks with Joint Route and Departure Time Choice. *Journal of the Transportation Research Board*, No 1923, pp 61-72.

56. Josefsson, M. and Patriksson,M. (2007). Sensitivity Analysis of Separable Traffic Equilibrium Equilibria with Application to Bilevel Optimization In Network Design. *Transportation Research Part B* 41, pp. 4-31

57. Karoonsoontawong, A. and Waller, S.T. (2005). A Comparison of System- and User-Optimal Stochastic Dynamic Network Design Models Using Monte Carlo Bounding Techniques. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1923, p 91-102.

58. Karoonsoontawong, A. (2006). Robustness Approach to the Integrated Network Design Problem, Signal Optimization and Dynamic Traffic Assignment Problem. *Ph.D. Dissertation*, The University of Texas at Austin, Austin, Texas.

59. Karoonsoontawong, A. and Waller, S.T. (2006). Dynamic Continuous Network Design Problem: Linear Bi-level Programming and Metaheuristic Approaches. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1964, pp. 104-117.

60. Karoonsoontawong, A. and Waller, S.T. (2007). Robust Dynamic Continuous Network Design Problem. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2029, pp 58-71.

61. Karoonsoontawong, A. and S.T. Waller. (2008). Integrated Network Capacity Expansion and Traffic Signal Optimization Problem: Robust Bi-level Dynamic Formulation. *Networks and Spatial Economics*, DOI 10.1007/s11067-008-9071-x.

62. Kleijnen, J. P. C. (1995). Verification and Validation of Simulation Models. *European Journal of Operational Research*, Vol. 82, pp. 145–162.

63. Knight, F.H. (1924). Some Fallacies in the Interpretation of Social Cost. *Quarterly Journal of Economics*, 38, pp. 582-606.

64. Kunde, K. K. (2002). Calibration of Mesoscopic Traffic Simulation Models for Dynamic Traffic Assignment., *Master Thesis*, Massachusetts Institute of Technology.

65. Labbe, M., Marcotte, P. and Savard, G. (1998). A Bilevel Model of Taxation and Its Application to Optimal Highway Pricing. *Management Science*, 44 (12), pp. 1608–1622.

66. Lawphongpanich,S. and Hearn, D.W. (2004). An MPEC Approach to Second Best Toll Pricing. *Mathematical Programming, Series B*, Vol. 101, No.1, pp.33-55.

67. LeBlanc, L. J., and D. Boyce. (1986). A Bi-Level Programming Algorithm for the Exact Solution of the Network Design Problem with User-Optimal Traffic Flows. *Transportation Research B*, Vol. 20, pp. 259-265.

68. LeBlanc, L. J. (1975). An Algorithm for the Discrete Network Design Problem. *Transportation Science*, Vol. 9, pp. 183-199.

69. LeBlanc, L. J., and M. Abdulaal. (1979). An Efficient Dual Approach to the Urban Road Network Design Problem. *Computers and Mathematics with Applications*, Vol. 5, pp. 11-19.

70. LeBlanc, L. J., and M. Abdulaal. (1984). A Comparison of the User-Optimum Versus System-Optimum Traffic Assignment in Transportation Network Design. *Transportation Research B*, Vol. 18, pp.115-121.

71. Li, Y., A. K. Ziliaskopoulos, and S.T. Waller. (1999) Linear Programming Formulations for System Optimum Dynamic Traffic Assignment with Arrival Time–Based and Departure Time–Based Demands, *Transportation Research Record: Journal of the Transportation Research Board*, No. 1667, 52-59.

72. Li, Y., Waller, S.T. and Ziliaskopoulos, T. (2003). A Decomposition Scheme for System Optimal Dynamic Traffic Assignment Models. *Transportation Research Record*, Vol. 3, No. 4, pp. 441-455.

73. Lighthill, M.J., Whitham, J.B. (1955). On Kinematic Waves II: A Theory of Traffic Flow on Long Crowded roads. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, Vol. 229, No. 1178 , pp. 317-345.

74. Lin, D.-Y., Karoonsoontawong, A. and Waller, S.T., (2009). A Dantzig-Wolfe Decomposition Based Heuristic Scheme for Bi-level Dynamic Network Design Problem. *Networks and Spatial Economics* (in press).

75. Lin, D.-Y., A. Unnikrishnan, and S.T. Waller. (2008) A Genetic Algorithm for Bi-level Linear Programming Dynamic Network Design Problem, *Technical Paper*, University of Texas at Austin.

76. Lundgren, J.T. and Peterson, A. (2008) A Heuristic for the Bilevel Origin-destination-matrix Estimation Problem. *Transportation Research Part B*, Vol. 42, Issue 4, pp. 339-354.

77. Magnanti, T. L., and R. T. Wong. (1984). Network Design and Transportation Planning: Models and Algorithms. *Transportation Science*, Vol. 18, No. 1, pp. 1-55.

78. Mahmassani, H.S., Zhou, X. and Lu, C.-C. (2005) Toll Pricing and Heterogeneous Users: Approximation Algorithms for Finding Bicriterion Time-Dependent Efficient Paths in Large-Scale Traffic Networks, *Transportation Research Record*, No. 1923, pp. 28–36.

79. Mahut, M., Florian, M., Tremblay, N., Campbell, M., Patman, D., and McDaniel, Z. K. (2004). Calibration and Application of a Simulation-Based Dynamic Traffic Assignment Model, *Transportation Research Record*, No. 1876, pp. 101–111.

80. Marcotte, P. (1983). Network Optimization with Continuous Control Parameters. *Transportation Science*, Vol. 17, pp. 181-197.

81. Marcotte, P. A (1988) Note on A Bilevel Programming Algorithm By Leblanc and Boyce. *Transportation Research Part B*, Vol. 22, 1988, pp. 233-237.

82. Meng, Q., H. Yang, and M. G. H. Bell. (2001). An Equivalent Continuously Differentiable Model and a Locally Convergent Algorithm for the Continuous Network Design Problem. *Transportation Research B*, Vol. 35, pp. 83-105.

83. Morrison, S.A. (1986). A Survey of Road Pricing. *Transportation Research*, 20A (2), pp. 87-97.

84. Muloz, L., Sun, X., Sun, D., Gomes, G. and Horowitz, R. (2004). Methodological Calibration of the Cell Transmission Model., *Proceeding of the 2004 American Control Conference*, pp. 798-803.

85. Murtagh, B. and Saunders, M. (1998) MINOS 5.5 User's Guide, Stanford University.

86. Patriksson, M., and R. T. Rockafellar. (2002). A Mathematical Model and Descent Algorithm for Bilevel Traffic Management. *Transportation Science*, Vol. 36, No. 3, pp. 271-291.

87. Peeta, S. and Ziliaskopoulos, A.K. (2001). Foundations of Dynamic Traffic Assignment: The Past, the present and the Future, *Networks and Spatial Economics*, Vol. 1, No.3-4, pp.233-265.

88. Peeta, S. and Yu, J.W. (2006). Behavior-Based Consistency-Seeking Models as Deployment Alternatives to Dynamic Traffic Assignment Models. *Transportation Research Part C*, Vol. 14, Issue 2, pp 114-138.

89. Policy and Economic Analysis Unit (2003). The Value-of-Travel Time: Estimates of the Hourly Value of Time for Vehicles in Oregon 2003. Oregon Department of Transportation.

90. Richards, P.I. (1956) Shockwaves on the highway. *Operations Research*, Vol. 4, 42-51.

91. Powell, M. J. (1964) An efficient method for finding the minimum of a function of several variables without using derivatives. *The Computer Journal*, Vol. 9, pp. 155-162.

92. Sherali, H.D., Arora, N. and Hobeika, A.G. (1997) Parameter Optimization Methods for Estimating Dynamic Origin-Destination Trip-Tables, *Transportation Research Part B*, Vol. 31, Issue 2, pp.141-157.

93. Sherali, H.D. and Park, T. (2001) Estimation of Dynamic Origin–Destination Trip Tables for a General Network. *Transportation Research Part B*, Vol. 35, Issue 3, pp.217-235.

94. Shor, P. W. (1994) Algorithms for quantum computation: Discrete logarithms and factoring. *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science*, 124-134.

95. Shor, P. W. (1997) Polynomial-time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, *SIAM Journal on Computing*, Vol. 26, No. 5, 1484-1509.

96. Steane, A. (1998) Quantum Computing. *Reports on Progress in Physics*, Vol. 61, No.2, 117-173.

97. Suwansirikul, C., T. L. Friesz, and R. L. Tobin. (1987). Equilibrium Decomposed Optimization: A Heuristic for the Continuous Equilibrium Network Design Problem. *Transportation Science*, Vol. 21, pp. 254-263.

98. Tobin, R.L. and Friesz, T.L., 1988. Sensitivity Analysis for Equilibrium Network Flow. *Transportation Science*, 22, pp.242–250.

99. Tony, H. (1999) Quantum Computing: An Introduction. *Computing and Control Engineering Journal*, 105-112.

100. Turner, S., Albert, L., Gajewski, B. and Eisele, W. (2000). Archived intelligent transportation system data quality: Preliminary analyses of San Antonio TransGuide data. *Transportation Research Record*, No.1719, pp. 77–84.

101. Uchida, K., Sumalee, A., Watling, D. and Connors, R. (2008) A Study on Network Design Problems for Multi-modal Networks by Probit-based Stochastic User Equilibrium. *Networks and Spatial Economics*, Vol. 7, No. 3, pp.213-240.

102. Ukkusuri, S. (2002). Linear Programs for the User Optimal Dynamic Traffic Assignment Problem. *Master Thesis*, University of Illinois at Urbana-Champaign.

103. Ukkusuri, S., Karoonsoontawong, A. and Waller, S. T. (2004). A Stochastic Dynamic User Optimal Network Design Model Accounting for Demand Uncertainty. *Proceedings of the International Conference of Transportations Systems Planning and Operations* (TRANSPO), Madras, India, Feb. 18-20.

104. Ukkusuri, S. and Waller, S. T. (2008). Linear Programming Models for the User and System Optimal Dynamic Network Design Problem: Formulations, Comparisons and Extensions. *Networks and Spatial Economics*, Vol.8, No. 4, pp.383-406.

105. Verhoef, E.T. (2002). Second-best Congestion Pricing in General Networks: Algorithms for Finding Second-best Optimal Toll Levels and Toll Points. *Transportation Research Part B*, 36(8), pp. 707–729.

106. Vicente, L. N. and Calamai, P.H. (1994). Bilevel and Multilevel Programming: A Bibliography Review. *Journal of Global Optimization*, Vol. 5, pp.291-306.

107. Vickrey, W.S. (1969). Congestion Theory and Transport Investment. *American Economic Review* 59 (Papers and Proceedings) pp. 251-260.

108. Von Stackelberg, H. (1952). *The Theory of the Market Economy*, Oxford University Press, Oxford.

109. Waller, S.T. and Ziliaskopoulos, A.K. (1998) "A Visual Interactive System for Transportation Algorithms" Presented at the 78th Annual Meeting of the Transportation Research Board, Washington, D.C.

110. Waller, S. T. (2000) Optimization and Control of Stochastic Dynamic Transportation Systems: Formulations, Solution Methodologies, and Computational Experience. *Ph.D. Dissertation*, Northwestern University.

111. Waller, S. T. and Ziliaskopoulos, A. K. (2001). Stochastic Dynamic Network Design Problem. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1771, TRB, National Research Council, Washington, D.C., pp.106-113.

112. Waller, S.T. and Ziliaskopoulos, A.K. (2006). A Combinatorial User Optimal Dynamic Traffic Assignment Algorithm. *Annals of Operations Research*, Vol. 144, No. 1.

113. Waller, S.T., Mouskos, K.C., Kamaryiannis, D. and Ziliaskopoulos, A.K. (2006). A Linear Model for Continuous Network Design Problem. *Computer-Aided Civil and Infrastructure Engineering*, Vol. 21, No. 5, July, p 334-345.

114. Walters, A.A. (1961). The Theory and Measurement of Private and Social Cost of Highway Congestion. *Econometrica*, 29, pp. 676-699.

115. Wie, B., and Tobin, R. L. (1998). Dynamic Congestion Pricing Models for General Traffic Networks. *Transportation Record Part B*, Vol. 32, No. 5, pp. 313–327.

116. Wie, B.-W. (2007). Dynamic Stackelberg Equilibrium Congestion Pricing. *Transportation Research Part C*, 15, pp. 154-174.

117. Xie, C. (2008) Evacuation Network Optimization: Models, Solution Methods and Applications, *Ph.D. Dissertation*, Cornell University.

118. Yang, H. and Lam, W.H.K. (1996). Optimal road tolls under conditions of queuing and congestion. *Transportation Research Part A*, 30(5), pp. 319–332.

119. Yang, H. and Huang, H.-J., 1997. Analysis of Time-varying Pricing of a Bottleneck with Elastic Demand using Optimal Control Theory. *Transportation Research Part B*, 31B (6), pp. 425–440.

120. Yang, H., and M. G. H. Bell. (1998). Models and Algorithms for Road Network Design: A Review and Some New Developments. *Transport Reviews*, Vol. 18, No. 3, pp. 257-278.

121. Yildrim, M.B. and Hearn, D.W., 2005. A First Best Toll Pricing/next Term Framework for Variable Demand Traffic Assignment Problems. *Transportation Research Part B*, 39, pp. 659-678.

122. Yin, Y., Madanat, S.M. and Lu, X.-Y. (2008) Robust Improvement Schemes for Road Networks under Demand Uncertainty, *European Journal of Operational Research*, doi:10.1016/j.ejor.2008.09.008.

123. Ziliaskopoulos, A. K. (2000). A Linear Programming Model for the Single Destination System Optimum Dynamic Traffic Assignment Problem. *Transportation Science*, Vol. 34, No. 1, pp. 37-49.

Vita

Dung-Ying Lin was born in Chiayi City in Taiwan on 22 December 1977, the son of Chi-Lung Lin and Chian-Ying Wang. He received his Bachelor degree of Business Administration from National Cheng-Kung University where he majored in Transportation and Communication Management in 2000. He subsequently proceeded to graduate study and received his Masters of Business Administration degree at the same institution. In 2005, he entered the Graduate School of The University of Texas and pursued a Ph.D. in Transportation Engineering.

Permanent address: No.216, Linsen W. Rd., Chiayi City 600, Taiwan

This dissertation was typed by Dung-Ying Lin.